Final Report for NASA GRANT NC 2-5269:

Integrated Computational System for Aerodynamic Steering and Visualization

Principal Investigator:
Professor Lambertus Hesselink
Phone: 650-723-4850
Fax : 650-725-3459
bert@kaos.stanford.edu

Departments of Electrical Engineering, Applied Physics, and Aeronautics and Astronautics
Stanford University

October 2, 1999

1 Introduction

In February of 1994, an effort from the Fluid Dynamics and Information Sciences Divisions at NASA Ames Research Center with McDonnel Douglas Aerospace Company and Stanford University was initiated to develop, demonstrate, validate and disseminate automated software for numerical aerodynamic simulation. The goal of the initiative was to develop a tri-discipline approach encompassing CFD, Intelligent Systems, and Automated Flow Feature Recognition to improve the utility of CFD in the design cycle. This approach would then be represented through an intelligent computational system which could accept an engineer's definition of a problem and construct an optimal and reliable CFD solution.

Stanford University's role focused on developing technologies that advance visualization capabilities for analysis of CFD data, extract specific flow features useful for the design process, and compare CFD data with experimental data.

During the years 1995-1997, Stanford University focused on developing techniques in the area of tensor visualization and flow feature extraction. Software libraries were created enabling feature extraction and exploration of tensor fields. As a proof of concept, a prototype system called the Integrated Computational System (ICS) was developed to demonstrate CFD design cycle.

The current research effort focuses on finding a quantitative comparison of general vector fields based on topological features. Since the method relies on topological information, grid matching and vector alignment is not needed in the comparison. This is often a problem with many data comparison techniques. In addition, since only topology based information is stored and compared for each field, there is a significant compression of information that enables large databases to be quickly searched. This report will (1) briefly review the technologies developed during 1995-1997 (2) describe current technologies in the area of comparison techniques, (4) describe the theory of our new method researched during the grant year (5) summarize a few of the results and finally (6) discuss work within the last 6 months that are direct extensions from the grant.

2 Review of Previous Work (1995-1997)

This section reviews the various visualization techniques and tools developed under the grant during the years 1995 through 1997 that aid in the analysis of CFD data. Since this section is only a review, past papers are cited or enclosed that elaborate on the various techniques.

The tools are general enough to be applied to any fluid data set and for the topology tools to a broader set of scientific data; however, for the purpose of this section one particular problem is focused upon: a hemispherical cylinder with an incidence angle of 19 degrees, an incoming Mach number of 1.2 and a Reynolds number of 445,000. The extraction of shock waves and vortex features will be reviewed along with colored textures to simulate oil flow combined with pressure sensitive paint. In addition, a novel method for tensor visualization

using topological decomposition will be discussed. The section is completed by discussing how these tools are combined into an Integrated Computational System (ICS).

2.1 Shock Wave Extraction

A shock surface represents a sudden change of fluid properties. Typically, a shock is witnessed when a body travels at transonic speeds. The flow adjusts to the body by abruptly changing its direction with consequent changes in pressure, density, and temperature. This abruptness is caused by a nonlinear response of flow quantities such as pressure and density to the presence of a body or as a result of initial conditions. We have devised an effective method for shock extraction by applying knowledge of compressibility theory. By computing Mach numbers from velocities that are in the direction of the pressure gradient, and then extracting a corresponding isosurface, shock surfaces are identified. Noise and false detected surfaces (found passing through the boundary layer) are removed by enforcing the Rankine-Hugoniot relation, and by utilizing information about the geometry. A complete description of the algorithm is included in the supplemental entitled: *Summary of Work on Shock Wave Feature Extraction in 3-D Datasets.* The feature extractor was written using custom developed libraries that were specifically designed to ease the implementation of future extractors. This implementation reduced the solver's code size, and provided management and visualization of the extracted feature. Figure 1 depicts the bow shock upstream of the hemispherical cylinder's nose, along with a second shock on the top surface induced by the expansion due to the high angle of attack.

2.2 Oil Flow Simulation

3-D separated flows play a significant role in aerodynamics because of the close relationship between separation and the existence of vortices, and because of the effort placed in trying to avoid/understand them. Vortices are important structures of the flow far from the body as they affect lift and drag. The high-angle-of-attack flow about a hemisphere cylinder is a classical example of flows with massive separation. The first step in visualizing 3-D separated flows consists of depicting the structure of the vector field near the body, for which an adequate description is inferred from the skin-friction field—i.e., the 2-D tangential velocity field one grid plane away from the surface of the body. This method mimics a visualization method that is commonly used by experimentalists, namely, the oil flow method.

A recent major improvement was made by adopting a new rendering technique, namely anisotropic textures. By generating textures, we render 2-D streamlines directly, without the need for integrating the vector field (Reference [1]). Textures provide an adequate solution to problems of accuracy, do not require an integration of the whole vector field, and create continuous images.

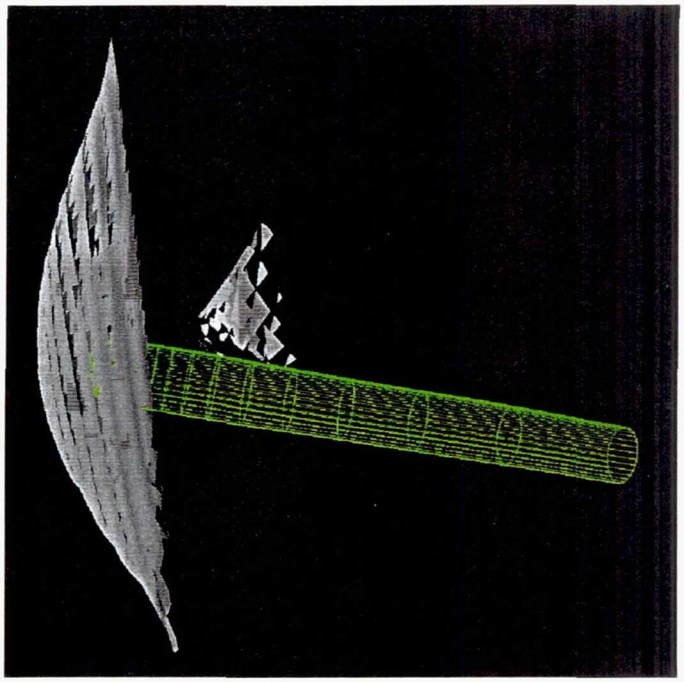


Figure 1: Shock extraction about a hemisphere cylinder.

By combining coloring based on pressure values on the body surface this technique of oil flow simulation provides the researcher with an image that can be analyzed using critical point theory and can be compared with images from experiments. Figure 2 contains such a description of the flow. From the image one can clearly identify the lines of separation and re-attachment along with the critical points on them. Details of the algorithm are described in the supplemental entitled: *Flow Visualization with Textures.*

2.3 Vortex Extraction

Vortex identification is an important flow feature in the design of aircrafts. Locating vortices can aid in improving aircraft safety, and airport landing and take-off throughput. It can also help increase turbo-machinery efficiencies and reduce airframe vibration and audible noise.

One method for vortex detection is to use topological skeletons. Separation lines can be located using critical point theory. These surfaces of separation and the associated vortices are extensions in the flow of the skin-friction topology. The surfaces of separation emanate from the body along lines of separation. They are computed by advecting a front of streamlines initially positioned along the lines of separation. Figure 3 shows the surfaces of separation which roll up

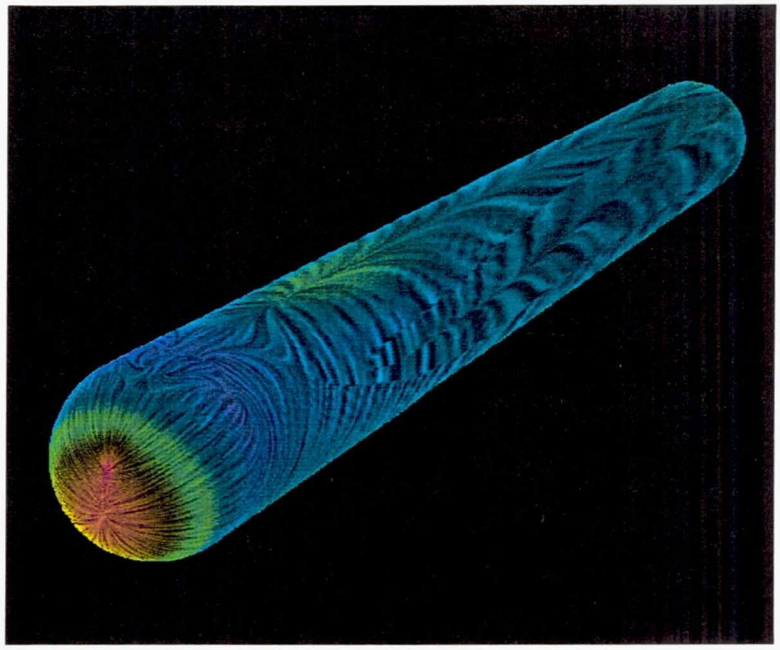


Figure 2: Oil flow simulation combined with pressure sensitive paint

to form the vortices.

2.3.1 Vortex Core Extraction

There are several prevalent techniques for the identification of vortex cores. Critical point theory [2], pressure/density minima and helicity are just a few techniques used to extract vortex cores. Observations relating the medium eigenvector of the deformation tensor to the vorticity vector [3, 4] have lead to investigations into techniques based on the topology of tensor fields and our idea of hyperstreamlines. Our method for vortex core extraction is based on using the helicity density gradient [5]. This method is effective in subsonic flows where density methods typically fail. Unlike methods that use enstrophy, the sign is preserved hence allowing the identification of secondary vortices via the swirl direction. This method also produces a continuous feature by integrating along the helicity density gradient vector, unlike methods based on critical point theory which use interpolation. Figure 4 depicts one of the four vortex cores (in blue), and the relation to the surrounding flow via hyperstreamlines of the reversible momentum tensor.

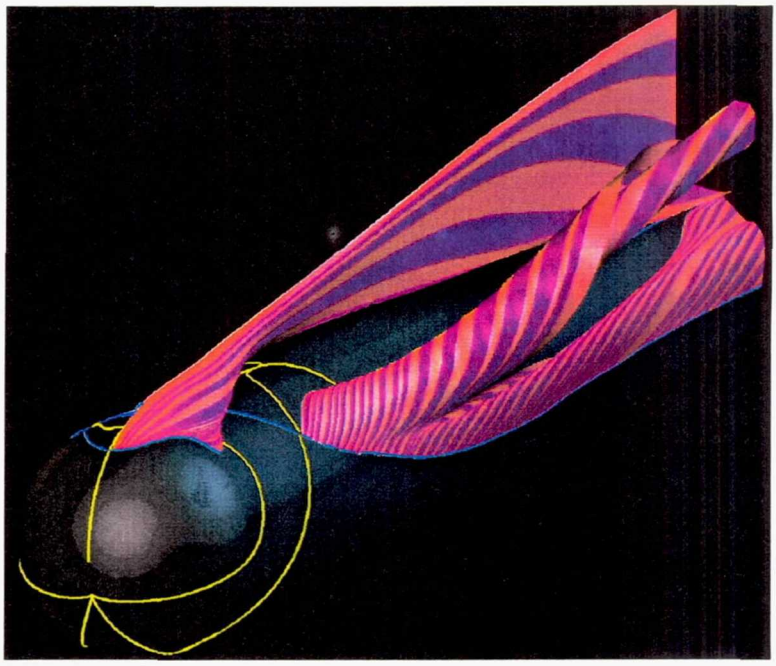


Figure 3: Stream surfaces depicting separation topology

2.4 Tensor Visualization

Tensor data sets are at the heart of many engineering and physics disciplines, yet few methods have been devised for understanding and visualizing such data. In particular, second-order tensor fields are central to fluid mechanics. For example stresses, strain rate, and Reynolds stresses are all tensor quantities. In traditional approaches, selected data are displayed using simple local icons depicting, for example, eigenvalues and eigenvectors at selected points. Through this data reduction process, valuable information is lost or discarded.

Since a tensor field is a continuous field, it should be visualized as such. The approach we have taken is to trace the trajectories of the eigenvectors of the tensor field (referred to as *hyperstreamlines* [6]). Any symmetric tensor data set can be represented locally by a set of orthogonal eigenvectors and eigenvalues which contain all the information about the tensor data at a given point in space. Hyperstreamlines are tangent curves to the principal eigenvectors; the principal eigenvectors are ordered according to the magnitude of their eigenvalues. After a particular eigenvector is chosen, integrating along the direction of the chosen eigenvector, a tangent curve is generated along which we can display, in an orthogonal manner, the remaining two eigenvectors. This can be done as a set of orthogonal vectors or ellipses having principal axes in proportion to the two eigenvalues.

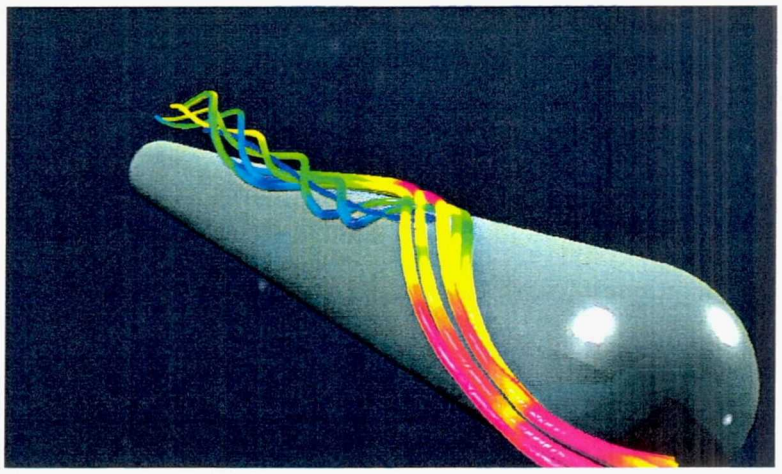


Figure 4: Vortex core (solid blue) encompassed by hyperstreamlines (major eigenvector) of the reversible momentum tensor

In this manner, a display is generated that takes advantage of the continuity of the data, and all information in the original data are retained. Figure 5 depicts the reversible momentum tensor in a clear and concise form. Hot colored regions depict kinetic energy density, while the hyperstreamlines trace the flow's path. Details of using tensor visualization methods in the CFD process are discussed in the following enclosed supplements: *The Topology of Symmetric Tensor Fields* [7], *The Topology of Three-Dimensional Symmetric Tensor Fields* [8], *The Topology of Symmetric, Second-Order 3-D Tensor Fields* [9].

2.5 Integrated Computational System (ICS)

Over the past few years, the development of the techniques described above were largely funded by NASA Ames with additional NSF funding for the work in visualizing the topology of 3-D tensor fields. To unify the visualization effort based on the principle *"Analyze then Visualize"*, a framework for solving flow features was designed recently using an object-oriented methodology. Exploiting encapsulation, inheritance, and dynamic function binding, we have built a comprehensive and extendible system, known as Integrated Computational System (ICS). The tools in ICS have been applied to solving problems in computational fluid and solid mechanics and can be used to compare results of multiple flow solutions. Schematically this system is shown in Figure 6.

Each feature extraction tool is a separate inherited solver. Each feature solver inherits data and methods from a base feature solver, and are all managed by a feature solver manager. Problems are submitted to the manager in a generalized, typically attribute oriented way. The feature solver manager makes the decision and returns a specific feature solver that is best suited for the

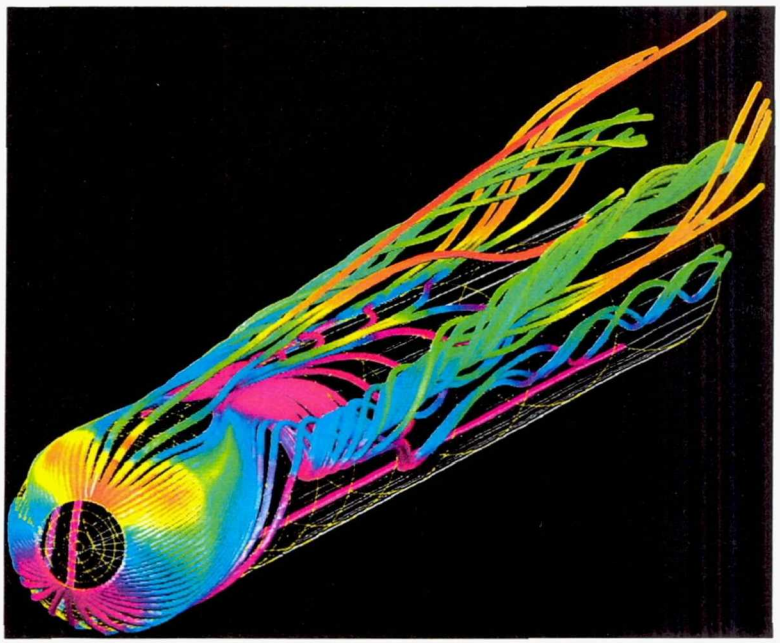


Figure 5: Rake of hyperstreamlines of the major eigenvector field of the reversible momentum tensor.

problem description. This returned solver being a child of a base feature solver has a common interface. This provides a great benefit in terms of minimizing documentation, and standardizing the interaction with other feature solvers that will be created. This polymorphic behavior has been extended to solutions as well. Each feature solver returns a solution which is a feature that has inherited from a base feature class, known as a primitive. A standard interface exists for a primitive to display itself to the screen, or in other formats such as VRML [1]. The primitive is typically small, and usually contains some compression such that the transfer to the client's machine requires a minimal amount of bandwidth. It is important to note that a feature solver manager provides a division between the interface and the implementation. The manager can be running on the same system as the client code or on another computational server in a different connected network.

The prototype currently provides the means to conduct a simulation using a full Navier Stokes flow solver, and allows automatic extraction of flow features such as shocks, vortex cores, and simulated oil flow combined with pressure sensitive paint. It also includes a tensor visualization module that creates tensor quantities relevant to fluid mechanics such as the velocity gradient, deformation,

[1] Virtual Reality Modeling Language provides a standard for three-dimensional data to be shared across the internet

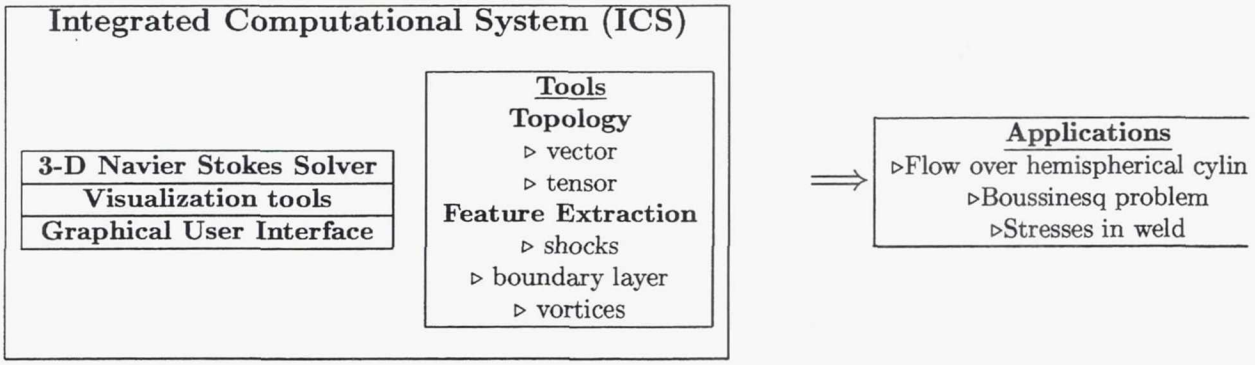


Figure 6: Visualization tools developed by our group at Stanford University.

and viscous stress tensors. In addition, user defined tensors can be loaded and explored within the system. All data is visualized using a custom developed application known as *FASTlook* which is similar to Silicon Graphic's Open Inventor's *ivview*. In contrast, however, *FASTlook* is designed for visualization of data-rich primitives found in scientific visualization. In addition, at the request of NASA Ames, *FASTlook* has been converted to a network helper application with the intention of incorporating the software within web based applications such as DARWIN [10].

Supplemental documentation is included that describes FASTlook entitled *PrimView: A Scientific Visualizer User's Guide v*2.0 and documentation describing the use of the solution primitives entitled *PrimView: A Scientific Visualizer Library Guide v*2.0.

3 Comparison Techniques

There exist a variety of comparison techniques for vector fields. These techniques basically fall into three general categories: Image, data, and feature extraction based comparisons. In most of these cases, comparisons are made visually [11]. Image based comparisons work on the computer generated image. Often times, a numerical data set is converted into an image that simulates an experimental visualization technique (computational flow imaging). This may be easier than extracting a vector field from an image, such as Schlieren. However, visualizing a field in 3-D is quite difficult. Often times, these techniques are limited to two dimensions. In addition to side-by-side comparison of images, other techniques include image fusion, and Fourier analysis [12].

Data level comparison techniques operate directly on the raw data. An accurate comparison requires proper grid alignment which can involve problematic interpolation between two fields [13].

The last comparison category is the extraction of features. Typically fea-

tures are flow specific such as vortex cores, shock surfaces, or topology. Often times there is a geometric representation of the feature and possibly a semantic representation of the system which can be compared using a pattern recognition technique [14]. This may lead to more robust comparisons. Past study in our group has focused on the geometric structure of vector fields [15]. However, this geometric structure can be visually deceiving since two vector fields may have the same underlying topological structure but are dissimilar in appearance [16]. Therefore, a quantitative measurement for comparison of vector fields is essential.

4 Description of a Vector Field

A 2-D vector field can be described as a system of two simultaneous differential equations having the following form:

$$\begin{aligned} v_x &= \frac{dx}{dt} = F(x,y) \\ v_y &= \frac{dy}{dt} = G(x,y) \end{aligned} \tag{1}$$

where F and G are continuous and have continuous partial derivatives in some region D.

A vector field is typically described by the number, type, and arrangement of critical points (or equilibrium points). These points are where the system is defined to be $F(x,y) = 0$, $G(x,y) = 0$. The number and nature of critical points will not change under continuous transformation. A critical point is said to be isolated or simple if there is an open neighborhood around it that contains no other critical points. For this report, we focus entirely on simple critical points. The global topology of the vector field is defined as the critical points and the set of their connecting streamlines. These streamlines (separatrices) divide the field into regions that are topologically equivalent to uniform flow. Hence, only the topology is needed to reconstruct the field and therefore is useful as a means of differentiating vector fields.

4.1 Classification of Simple Critical Points

The behavior of the flow about a critical point can be analyzed by investigating the streamlines in the neighborhood of the critical point. If we are sufficiently close to the critical point (say a distance dx,dy away) in most cases a first order Taylor series expansion of the velocity field is sufficient:

$$\begin{aligned} v_x(dx,dy) &\approx \frac{\partial v_x}{\partial x}dx + \frac{\partial v_x}{\partial y}dy \\ v_y(dx,dy) &\approx \frac{\partial v_y}{\partial x}dx + \frac{\partial v_y}{\partial y}dy \end{aligned} \tag{2}$$

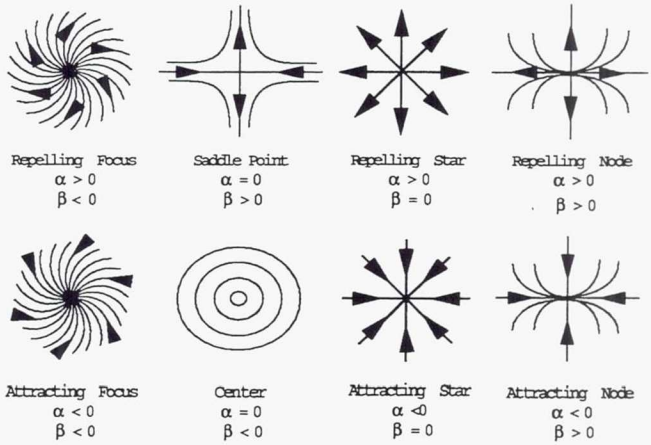


Figure 7: basic patterns for simple critical points

Hence, the flow pattern is completely determined by the $2X2$ Jacobian matrix, $J_{ij} = \frac{\partial v_i}{\partial v_j}$ $(i, j = 1, 2)$ evaluated at the critical point location. The various patterns formed in the phase-plane space can be seen by analyzing the eigenvalues of the Jacobian. The patterns are sketched in Figure 7. Notice a positive or negative real part (denoted by α) is indicative of repelling/attracting behavior. And if an eigenvalue has an imaginary part ($\beta < 0$), it indicates circulation about the point, otherwise asymptotic behavior is exhibited.

5 Vector Field Representation using Clifford Algebra

In [17] [18], Sheuermann et al. introduced Clifford algebra for vector field visualization. Clifford algebra provides a nice way to describe the relation between real and complex numbers in $2D$ space. The vector fields are defined over a complex field in this algebra and the nonlinear vector fields are represented as multiplications of linear fields.

For the Euclidean plane we get a 4-dimensional R-algebra G_2 with the basis $1, e_1, e_2, i = e_1 e_2$ as a real vector space. Multiplication is defined as associative, bilinear and by the equations

$$1e_j = e_j, j = 1, 2 \quad (3)$$

$$e_i e_j = 1, j = 1, 2 \quad (4)$$

$$i = e_1 e_2 = e_2 e_1 \quad (5)$$

$$1^2 = 1, j = 1, 2 \quad (6)$$

$$e_j^2 = 1, j = 1, 2 \quad (7)$$

$$i^2 = -1 \tag{8}$$

with

$$1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad i = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad e_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad e_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The usual vectors $(x, y) \in R^2$ are identified with

$$xe_1 + ye_2 \in E^2 \subset G_2 \tag{9}$$

and the complex numbers $a + bi \in C$ with

$$a1 + bi \in G_2 \tag{10}$$

5.1 Vector fields in Clifford space

A Clifford vector field is just a multivector field with values in $R^2 \subset G_2$

$$v : R^2 \to R^2 \subset G_2 \tag{11}$$

Let $z = x + iy$, $\bar{z} = x - iy$ be complex numbers in the Clifford algebra. This means

$$x = \frac{1}{2}(z + \bar{z}) \tag{12}$$

$$y = \frac{1}{2i}(z - \bar{z}) \tag{13}$$

We get

$$\begin{aligned} \vec{v}(r) &= v_1(x,y)e_1 + v_2(x,y)e_2 \\ &= v_1\left(\frac{1}{2}(z+\bar{z}), \frac{1}{2i}(z-\bar{z})\right)e_1 \\ &\quad -iv_2\left(\frac{1}{2}(z+\bar{z}), \frac{1}{2i}(z-\bar{z})\right)e_1 \\ &= E(z,\bar{z})\,e_1 \end{aligned} \tag{14}$$

Generally, a linear vector field can easily be shown as:

$$\begin{aligned} \vec{v}(r) &= E(z,\bar{z})e_1 \\ &= (az + b\bar{z} + c)e_1 \end{aligned} \tag{15}$$

where $a, b, c \subset C$.

Let $E : C^2 \to C$ be the polynomial so that $\vec{v} = E(z,\bar{z})e_1$. Let $F_k : C^2 \to C, k = 1, \ldots, n$ be the irreducible components of E, so that $E(z,\bar{z}) = \prod_{k=1}^{n} F_k$,

Table 1: Classification of Critical Points using α β values

α	β	Type	α	β	Type
$=0$	<0	Center	—	>0	Saddle $\lvert\beta\rvert > \lvert\alpha\rvert$
>0	<0	Repelling Focus	<0	<0	Attracting Focus
>0	$=0$	Repelling Star	<0	$=0$	Attracting Star
>0	>0	Repelling Node $\lvert\beta\rvert < \lvert\alpha\rvert$	<0	>0	Attracting Node $\lvert\beta\rvert < \lvert\alpha\rvert$

then an arbitrary polynomial vector field with isolated critical points can be expressed as:

$$\begin{aligned} \vec{v}(r) &= E(z,\bar{z})e_1 \\ &= \prod_{k=1}^{n}(a_k z + b_k \bar{z} + c_k)e_1 \end{aligned} \tag{16}$$

where z_k is the unique zero of $a_k z + b_k \bar{z} + c_k$.

6 α-β Space and its Use as a Metric

For a linear vector field $\vec{v} = (az + b\bar{z} + c)e_1$, let $a = a_1 + a_2 i$ and $b = b_1 + b_2 i$. Eigenvalues of the Jacobian around its critical point z_0 are $\lambda_1 = b_1 + \sqrt{|a|^2 - b_2^2}$ and $\lambda_2 = b_1 - \sqrt{|a|^2 - b_2^2}$.

Let $\alpha = b_1$ and $\beta = sign(|a|^2 - b_2^2)\sqrt{||a|^2 - b_2^2|}$, criteria for basic patterns of simple critical points are:

Selection of α and β as shown in Figure 7 and delineated below can be mapped to a_1, b_1, a_2, b_2 to yield any desired field:

Notice our definition of saddle is more relaxed than shown in Figure 7. The values of α and β determine the type of critical point but it is not sufficient to be used as a metric to differentiate between two types of critical points. So we introduce a new α-β space where the 8 simple critical points are mapped onto the α,β axes at their respective (α, β) points. Vectors in this space obey all the rules defined for a regular 2-D Euclidean space. All points in this space are normalized as follows:

$$\alpha' = \frac{\alpha}{\sqrt{\alpha^2+\beta^2}} \qquad \beta' = \frac{\beta}{\sqrt{\alpha^2+\beta^2}} \tag{17}$$

It is shown in [19] that the actual values of α and β do not determine the portrait of the critical point only the ratio between them. Hence, this normalization maps all points onto a unit circle (Figure 8) and thereby provides a means of relatively quantifying the difference between various points. Also note that a regular vector field with no critical points, $\vec{v} = const \cdot e_1$ has $\alpha = 0$ and $\beta = 0$ and sits at the origin of the unit circle. For the remainder of the report, α and β values will be assumed normalized.

A multiple point with a set of α's and β's corresponds to a set of points in the $\alpha - \beta$ space. For example, $\vec{v} = \bar{z}^2 e_1$ is a dipole which has two $(1, 0)$ point

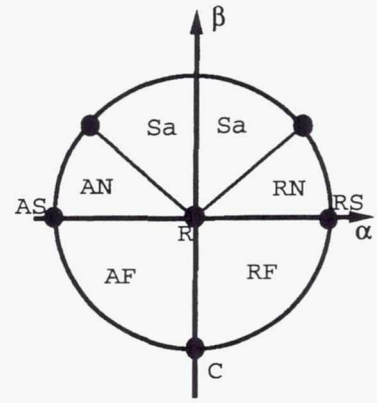


Figure 8: Basic patterns for critical points in α-β space; C for center, RN for node, AN for attracting node, RF for repelling focus, AF for attracting focus, St for star, Sa for Saddle and R for regular point

in $\alpha - \beta$ space; and $\vec{v} = (z - (2+2i)\bar{z} + c_1)(z + (2+2i)\bar{z} + c_2)e_1$ has one point at $(-\frac{2}{\sqrt{7}}, -\frac{\sqrt{3}}{\sqrt{7}})$ and another point at $(\frac{2}{\sqrt{7}}, \frac{\sqrt{3}}{\sqrt{7}})$ in $\alpha - \beta$ space.

7 Earth Mover's Distance

7.1 EMD analysis

The Earth Mover's Distance is first introduced in [20] [21] for content-based image retrieval in a large data base. It is used to compute the minimal amount of work that must be performed to transform one feature distribution into the other. Feature distribution in [20] [21] are the color and texture signatures of an image.

After careful study, we found that the EMD concept can be used to compute the differences between vector fields. Here, the feature distribution is redefined as the characteristics of a vector field.

Definition 1 (feature distribution) *A feature distribution for a vector field is the set of α and β values associated with the vector field's critical points:*

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}.$$

Definition 2 (Energy) *The energy for a vector field is:*

$$Energy = \sqrt{\sum_{i=1}^{n} (\alpha_i^2 + \beta_i^2)},$$

where n is the total number of critical points in this field.

This energy here is a quantity that characterizes the critical points of a vector field. It is different from the physical energy. The concept "work" is used to measure the energy differences between two vector fields or the amount of energy used to transform one vector field into the other.

Definition 3 (Work) *For two vector fields with feature distributions*

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}$$

and

$$\{(\alpha'_1, \beta'_1), (\alpha'_2, \beta'_2), \ldots, (\alpha'_n, \beta'_n)\}.$$

The amount of work necessary for transforming one vector field into the other is defined as: $Work = \sqrt{\sum_{i=1}^{n}((\alpha_i - \alpha'_i)^2 + (\beta_i - \beta'_i)^2)}$.

Intuitively, given two feature distributions, one distribution can be seen as a set of discrete point-objects with a certain amount of mass of earth spread in space, the other as a collection of holes in the same space. The work measures the least amount of energy needed to fill the holes with earth and is called the Earth Mover's Distance (EMD). Computing the EMD is based on a solution to the old *transportation problem* from linear optimization [22]. This is a bipartite network flow problem which can be formalized as the following linear programming problem: Let I be a set of suppliers, J a set of consumers, c_{ij} the cost to ship a unit of supply from $i \in I$ to $j \in J$

$$c_{ij} = \sqrt{(\alpha_i - \alpha_j)^2 + (\beta_i - \beta_j)^2}$$

and it is the same as the Euclidean distance $d_{ij} = \|\vec{v}_i - \vec{v}_j\|$ in α-β space. A critical point either exists as a whole or does not exist, it can not be split. In this case, the transportation problem has the property that the optimal flow f_{ij} can only be 0 or 1 [23]. We want to seek a set of f_{ij} that minimizes the overall cost:

$$EMD(\mathbf{x}, \mathbf{y}) = min \sum_{i \in I} \sum_{j \in J} c_{ij} f_{ij} \tag{18}$$

subject to the following constraints:

$$f_{ij} \geq 0 \quad i \in I, \; j \in J \tag{19}$$

$$\sum_{i \in I} f_{ij} = y_i, \quad j \in J \tag{20}$$

$$\sum_{j \in J} f_{ij} = x_i, \quad i \in I \tag{21}$$

$$\sum_{j \in J} y_j = \sum_{i \in I} x_i \tag{22}$$

Where x_i is the total supply of supplier i and y_j is the total capacity of consumer j. Constraint (19) allows shipping of supplies from a supplier to a consumer and not vice versa. Constraint (20) forces the consumers to fill up all of their capacities and constraint (21) limits the supply that a supplier can send as a total amount. Constraint (22) is a feasibility condition that ensures that the total demand equals the total supply, in other words, the distributions have the same overall mass and the EMD is a true metric [20].

It is likely that a set of vector fields will not have the same number of distributions. In order to satisfy constraint (22), we can create regular points to make the supply equal the demand without changing the vector fields. For example, if the supplier field contains 3 critical points

$$\vec{v} = \prod_{i=1}^{3}(a_i z + b_i \bar{z} + c_i)e_1 \tag{23}$$

and the consumer field contains 5 critical points

$$\vec{v'} = \prod_{j=1}^{5}(a'_i z + b'_i \bar{z} + c'_i)e_1 \tag{24}$$

The supplier side has two fewer points in the $\alpha - \beta$ space. Now let

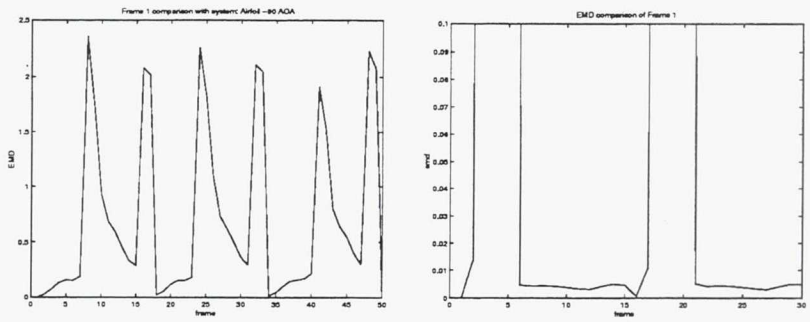
$$\vec{v} = \prod_{i=1}^{3}(a_i z + b_i \bar{z} + c_i) \cdot 1 \cdot 1e_1 \tag{25}$$

and the vector field remains unchanged. However, now we have two more regular points corresponding to 1 with $\alpha = 0$ and $\beta = 0$, and both the supplier and the consumer have 5 points in their feature distributions. All the conditions are satisfied, and we are ready to compute the EMD for these two fields and find out the dissimilarity between them.

In order to evaluate the meaningfulness of our new metric, we use *Multidimensional Scaling*(MDS) [24] [25] to embed the vector fields in a two-dimensional Euclidean space so that distances in the embedding are as close as possible to the true EMDs between vector fields. The MDS is introduced in the next section.

8 Display of EMDs for a Large Set of Vector Fields

The above discussions are for comparison of a pair of vector fields. If there exist a large set of vector fields and we want to compare their topologies, it is necessary to display them in a more meaningful way than a 1D list sorted by their EMDs. Yossi Rubner et al. have used Multidimensional Scaling Method (MDS) [20, 21] to display a set of images on a 2D map. Given n objects in a high dimension, the MDS method computes a configuration in a lower dimension space such that the distance between every pair of objects in this low dimension



a b

Figure 9: EMD of the flow over an a) airfoil b)circular cylinder.

space best matches the real distance in the high dimension. Inspired by their work, we compute the EMDs between every pair of vector fields and position the vector fields on a 2D map such that the distances between the vector fields match their EMD values as close as possible.

9 Application: Flow over an Airfoil and Cylinder

Rogers and Kwak computed the flow past a 2-D airfoil at $-90°$ angle of attack [26]. The model was of interest since the flow of the wake of an XV-15 Tilt Rotor aircraft degraded the lifting capability during hover. An incompressible, time accurate, Navier-Stokes code with artificial compressibility at a Reynolds number of 200 was used to compute the flow over a NACA 64A223M airfoil. Fifty frames were computed. During this time the flow entered into a periodic vortex shedding cycle. Earth mover's distance was computed over the 50 frames. The plot in Figure 9a depicts the EMD comparison of frame 1 with the remaining 49 frames. At frame 1, the EMD is zero, which is expected since the work required to convert a frame into itself is zero. The periodic nature is apparent. We see a repetition approximately every 17 frames. Also we see a sudden EMD rise when comparing frame 1 with frame 8 indicating a significant topological feature difference. Frame 1 contains three critical points: an attracting/repelling focus and a saddle. Frame 8 contains 5 critical points: two saddles, an attracting/repelling focus pair and a node. Since the flow is incompressible, the velocity divergence, $\nabla \cdot \vec{v}$, is expected to be zero everywhere in the flow. Hence only saddles and centers are to be extracted. However, it is common due to numerical computation for α to not exactly be zero, however, we should not expect to find a node. Upon closer examination of the data, the velocity divergence for certain frames is not zero near the tips of the foil where nodes are being extracted. We believe that the flow solver may not have fully

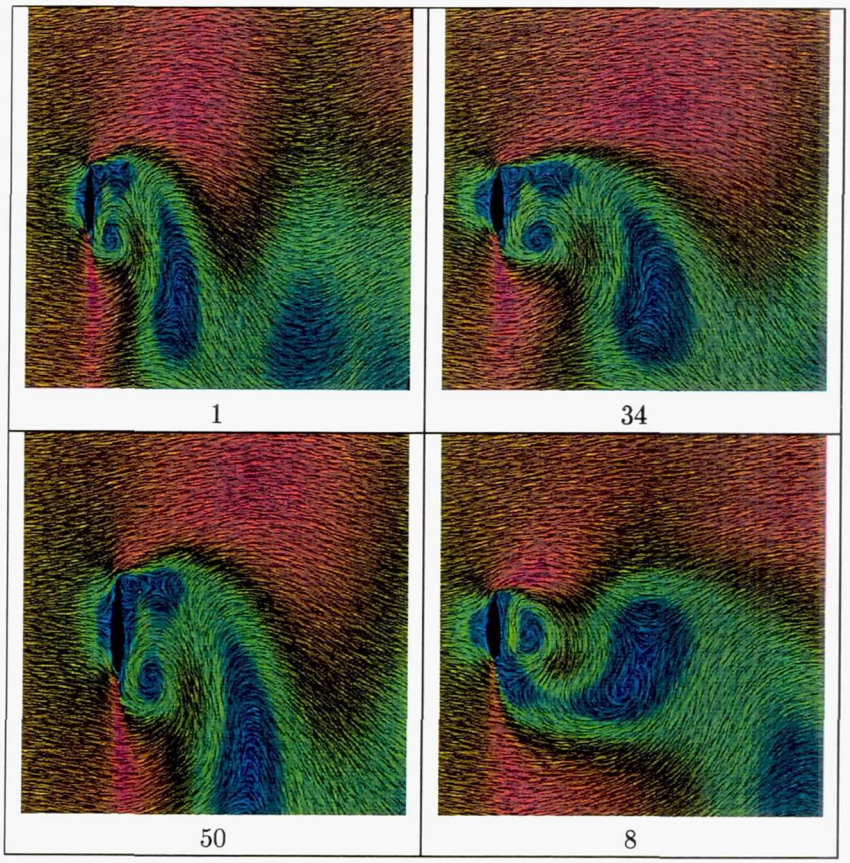


Figure 10: Topologically similar frames 1, 34 and 50 of flow about an airfoil. Frame 8 is topologically dissimilar.

converged and therefore we see this sudden jump of discontinuity in the field. The LIC images of frames 1, 34, and 50 are depicted in Figure 10 have very similar earth mover's distance and as can be seen look nearly identical. Frame 8 differs from the others due to its variation in topology (formation of nodes) and is apparent in the figure.

We contrast the flow over an airfoil with the flow over a circular cylinder simulated by Rogers and Kwak under the same flow conditions [26]. In this case, the flow is divergence free and the EMD values are quite similar. Thirty frames were computed capturing a complete cycle of vortex shedding. As can be seen from the plot in figure 9b, Frames 1 and 16 have nearly identical EMD values leading one to believe the period to be every 15 frames. Due to the symmetry of the flow, this is not far from the truth. In fact the flow produces a mirror image of itself every 15 frames as it sheds the alternate vortex and hence leads to the same topology. Figure 11b depicts the alternate vortex being shed to the image

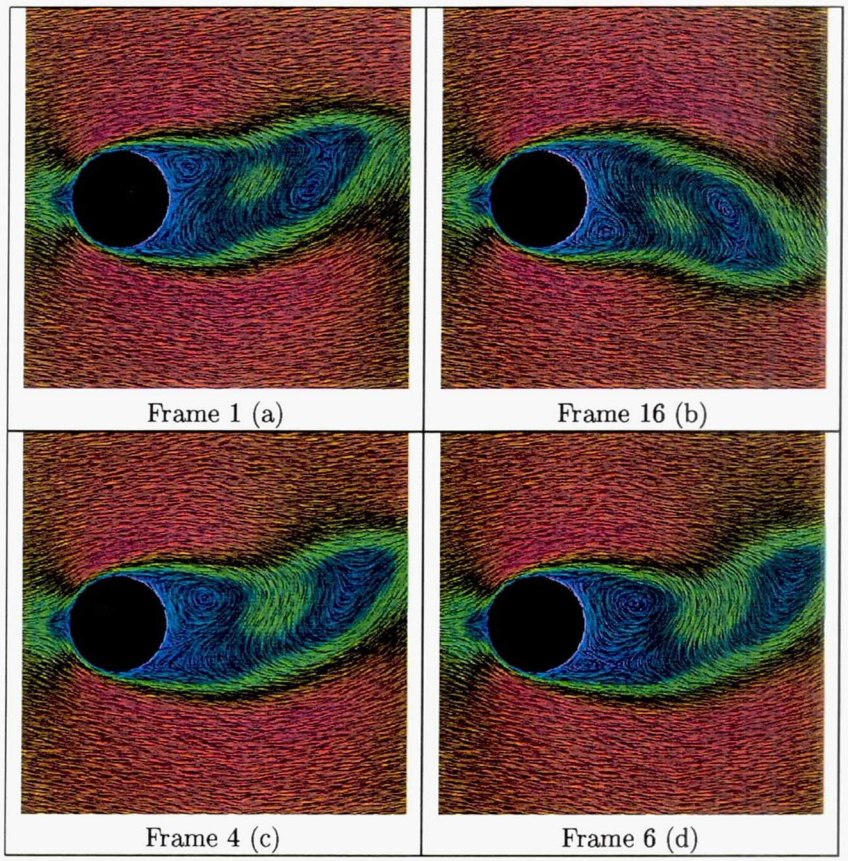


Figure 11: Topologically similar frames 1 and 16. Frame 4 depicts the down stream dissipation of saddle-center pair producing a larger EMD. Frame 6 depicts formation of new saddle-center pair at the bottom of the cylinder.

found in figure 11a. Furthermore, we see from figure 9b an increase in EMD value for frame 3. This increase is due to the dissipation of the saddle-center pair as it moves down stream (figure 11c). The EMD drops by frame 6 as the next saddle-center pair is shed (figure 11d). By frame 16, the saddle-center has moved down stream such that the α, β values are nearly identical to frame 1. Two frames later the saddle-center dissipate and the cycle repeats.

10 Discussion

We have demonstrated the effectiveness of topology based feature comparisons for vector fields. The use of a quantitative measure between fields provides the means for fast automated comparisons as well as an indepth study of flow fields

as demonstrated with the time history data. For the airfoil data, we have shown the effectiveness of the method as a diagnostic tool. The clear EMD difference provides an immediate alert into calculation problems for particular frames. For the cylindrical data, the periodic nature of the flow was revealed. The EMD difference also provides insight into the evolution of the flow field.

11 Direct Extensions from the Grant Research

The work conducted under the grant introduced a quantitative method for comparing 2-D vector fields based on the number and type of critical points that comprised the field. However, the arrangement of the critical points was not considered, potentially causing two very different fields with the same type of critical points but different streamline connections to be detected as similar fields. The work has been extended by considering the connections between critical points, thereby improving the representation of the vector field. The vector field's topology is represented using an attributed relational graph, and with the use of conventional graph matching algorithms a comparison of the fields can be made. The supplemental paper entitled *Topology Based Vector Field Comparisons Using Graph Methods* [27] discusses these graph methods.

Furthermore, the work under the grant focused on 2-D simple critical points. Many problems occur in three dimensions, and therefore a technique for 3-D comparisons has been recently developed. The extension to three-dimensions follows the path of our previous work, rethinking the representation of a critical point signature and the distance measure between the points. The discussion of this technique is included in the supplemental entitled *Feature Comparisons Of 3-D Vector Fields Using Earth Mover's Distance* [28].

References

[1] J. J. van Wijk, "Spot noise. Texture synthesis for data visualization," *Computer Graphics*, vol. 25, no. 4, pp. 309–318, 1991. NASA Langeley, Hampton VA.

[2] D. Kenwright and R. Haimes, "Vortex identification - applications in aerodynamics: A case study," in *Proc. IEEE/ACM Visualization '97*, pp. 413–415, Phoenix, Arizona., October 1997.

[3] J. Jimenez, "Kinematic alignment effects in turbulent flows," *Physics of Fluids*, vol. 4, pp. 652–654, April 1992.

[4] W. T. Ashurst, A. Kerstein, R. Kerr, and C. Gibson, "Alignment of vorticity and scalar gradient with strain rate in simulated navier-stokes turbulence," *Physics of Fluids*, no. 8, pp. 2343–2353, 1987.

[5] Y. Levy, D. Degani, and A. Seginer, "Graphical visualization of vortical flows by means of helicity," *AIAA Journal*, vol. 28, no. 8, pp. 1347–1352, 1990.

[6] T. Delmarcelle and L. Hesselink, "Visualizing second-order tensor fields with hyperstreamlines," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25–33, 1993.

[7] Y. Lavin, Y. Levy, R. Batra, and L. Hesselink, "The topology of symmetric tensor fields," in *13th AIAA Computational Fluid Dynamics Conference*, AIAA Snowmass, OctoberColorado, June 1997.

[8] Y. Lavin, Y. Levy, and L. Hesselink, "The topology of three-dimensional symmetric tensor fields," in *Late Breaking Hot Topics IEEE Visualization '96*, pp. 43–46, CS Press, Los Alamitos, CA., 1996.

[9] L. Hesselink, Y. Levy, and Y. Lavin, "The topology of symmetric, second-order 3-d tensor fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 1, pp. 1–11, 1997.

[10] J. Walton, D. Korsmeyer, R.K.Batra, and Y.Levy, "The darwin workspace environment for remote access to aeronautics data," in *AIAA 35th Aerospace Sciences Meeting & Exhibit*, no. 97-0667 in AIAA Paper, January 1997.

[11] F. H. Post and J. J. van Wijk, "Visual representations of vector fields: recent developments and research directions," in *Scientific Visualization* (L. Rosenblum, R. Earnshaw, and et al., eds.), pp. 181–195, Academic Press, Heidelberg, 1994.

[12] H.-G. Pagendarm and F. H. Post, "Comparative visualization-approaches and examples," in *Visualization in Scientific Computing* (M. Göbel, H. Müller, and B. Urban, eds.), pp. 95–108, Springer-Verlag, 1995.

[13] Q. Shen, A. Pang, and S. Uselton, "Data level comparison of wind tunnel and computational fluid dynamics data," in *Proc. IEEE Visualization '97*, pp. 67–74, CS Press, Los Alamitos, CA., 1997.

[14] L. Hesselink, "Digital image processing in flow visualization," *Annual Review of Fluid Mechanics*, vol. 20, pp. 421–485, 1988.

[15] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *IEEE Computer*, vol. 22, pp. 27–36, Aug 1989. Also appears in *Visualization in Scientific Computing*, G. M. Nielson & B. Shriver, eds. Companion videotape available from IEEE Computer Society Press.

[16] Y. Lavin, R. Batra, and L. Hesselink, "Feature comparisons of vector fields using earth mover's distance," in *Proc. IEEE/ACM Visualization '98*, pp. 413–415, North Carolina, October 1998.

[17] G.Sheuermann, H.Krüger, M.Menzel, and A.Rockwood, "Visualization of higher order singularities in vector fields," in *Proc. IEEE Visualization '97*, pp. 67–74, CS Press, Los Alamitos, CA., 1997.

[18] G.Sheuermann, H.Kruger, M.Menzel, and A.Rockwood, "Visualizing non-linear vector field topology," *Computer Graphics and Its Applications, Special Edition*, 1998.

[19] Y. Lavin, *Topology based visualization for vector and tensor fields.* PhD thesis, Stanford University, 1998.

[20] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. IEEE Internations Conferences on Computer Vision*, 1998.

[21] Y. Rubner, L. J. Guibas, and C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," in *Proceedings of the ARPA Image Understanding Workshop*, 1997.

[22] G. B. Dantzig, "Application of the simplex method to a transportation problem.," *Activity Analysis of Production and Allocation*, pp. 359–373, 1951.

[23] G. B. Dantzig, *Linear programming.* Springer series in operations research, Springer, 1997.

[24] J. Kruskal, "Multi-dimensional scaling by optimizing goodness-of-fit to a non-metric hypothesis.," *Psychometrika*, vol. 29, pp. 1–27, 1964.

[25] R. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function, i and ii.," *Psychometrika*, vol. 29, pp. 125–140,219–246, 1962.

[26] S. Rogers and D. Kwak, "An upwind differencing scheme for the time-accurate incompressible navier-stokes equations," in *Proceedings of the AIAA 6th Applied Aerodynamics Conference*, pp. 492–502, American Institute of Aeronautics and Astronautics, 1988.

[27] R. Batra, K. Kling, and L. Hesselink, "Topology based vector field comparisons using graph methods," in *Late Breaking Hot Topics IEEE Visualization '99*, p. To be published, CS Press, Los Alamitos, CA., 1999.

[28] R. Batra and L. Hesselink, "Feature comparisons of 3-d vector fields using earth mover's distance," in *Proc. IEEE Visualization '99*, p. To be published, IEEE Computer Society Press, Los Alamitos, CA, 1999.

TOPOLOGY BASED VISUALIZATION FOR VECTOR AND SECOND-ORDER TENSOR FIELDS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF PHYSICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Yingmei Lavin

August 1998

© Copyright by Yingmei Lavin 1998

All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Lambertus Hesselink
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Marc Levoy

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Sebastian Doniach

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

Scientific visualization transforms complex data obtained from experiments, observations and numerical simulations into an image amenable to understanding by the human visual system while maintaining the integrity of the information. It is a means to communicate between human perception and abstract physical world.

Vector (first-order tensor) and second-order tensor fields are multivariate, multidimensional data sets that are typically very difficult to comprehend. Visualizing such complex data is a very important subfield in scientific visualization.

Much research has been accomplished on vector field visualization. However, virtually no work has been carried out on quantitative comparisons of similarities and differences between vector fields. This dissertation introduces a novel approach to define a topology based measurement for such a purpose. The usefulness of this measurement can be seen when comparing computational and experimental flow fields under the same conditions. Furthermore, its applicability can be extended to such cumbersome tasks as navigating through a large data base by searching for a similar topology.

This new measure relies on the use of critical points, which are a key feature of vector field topology. In order to characterize critical points, α and β parameters are introduced. They are used to form a closed set of eight unique patterns for simple critical points. These patterns are also basic building blocks for higher order nonlinear vector fields. In order to study and compare a given set of vector fields, a measure of

distance between different patterns of critical points is introduced. The basic patterns of critical points are mapped onto a unit circle in α - β space. The concept of Earth Mover's Distance is used to compute the closeness between various pairs of vector fields, and a nearest-neighbor query is thus produced to illustrate the relationship between the given set of vector fields.

Very few methods have been developed to understand and visualize second-order tensor fields due to their complex nature and large data sizes. Topology based study has been done in two-dimensional space. This dissertation extends the research into three dimensions, an area that hasn't been studied before to the best of our knowledge.

The basic constituents of second-order tensor fields are degenerate points. They play a role similar to critical points in vector topology. In this dissertation, we address the conditions for the existence of degenerate points and based on these conditions we predict the distribution of degenerate points inside the field. Every tensor can be decomposed into a deviator and an isotropic tensor. A deviator determines the properties of a tensor field, while the isotropic part provides a uniform bias. Deviators can be three-dimensional or locally two-dimensional. The triple degenerate points of a tensor field are associated with the singular points of its deviator and the double degenerate points of a tensor field have singular local 2-D deviators. Control functions are in charge of the occurrences of a singularity of a deviator. These singularities can further be linked to important physical properties of the underlying physical phenomena. For example, for a deformation tensor in a stationary flow, the singularities of its deviator actually represent the area of the vortex core in the field; for a stress tensor, the singularities represent the area with no stress; for a viscous flow, removing the large, isotropic pressure contribution enhances dramatically the anisotropy due to viscosity.

Acknowledgements

I would like to thank Prof. Lambertus Hesselink for his advice and for providing an environment that helped me develop creativity and independence and for providing the support and the resources necessary for my research. I want to thank Dr. Yuval Levy for his insight on Fluid Mechanics, for all the useful discussions on both the theory and software sides and for his guidance in the early stage of my research; and Rajesh Batra for his critical comments on my papers and thesis, for all the useful discussions and constant help on both software and hardware sides. Also, I would like to thank everyone in the group for providing a pleasant working environment.

I want to express my deepest gratitude to my parents for raising me and putting great effort on my education; and my husband James for his constant encouragement and support.

Finally, I would like to thank Greg Zilliac from NASA Ames for providing various data sets of turbulence flows. My research was supported by NASA Ames Numerical Aerodynamics Simulation program, NASA Ames Fluid Dynamics Division, NASA Contract NAG 2-911 and NASA grant ECS-9215145.

Contents

Abstract iv

Acknowledgements vi

1 Introduction 1

1.1 Scientific Visualization 1

1.2 Vector and Second-order Tensor Fields 2

1.2.1 Vectors, Second-order Tensors and the Field Concept 3

1.2.2 Combinatorial Topology of Vector and Tensor Fields 6

1.3 Motivations and Objectives 7

1.3.1 Overview of vector field visualization 7

1.3.2 Overview of tensor field visualization 8

1.3.3 Objectives 9

1.4 Visualization Techniques for Vector and Tensor Fields 13

1.4.1 Vector Fields 14

1.4.2 Tensor Fields 21

1.4.3 Texture Mappings 26

1.5 Structure of the Dissertation 28

2 Review of Vector and Tensor Field Topology 30

2.1 Vector Field Topology 31
2.1.1 Critical Point Theory 32
2.1.2 Global Topology 39
2.2 Tensor Field Topology 44
2.2.1 Degenerate Point Theory 46
2.2.2 Global Topology 57
2.3 Chapter Summary 61

3 Feature comparisons of vector fields using Earth Mover's Distance 63
3.1 Clifford Algebra 64
3.2 Vector field representations using Clifford Algebra 66
3.3 α-β space analysis 68
3.3.1 Introduction to α-β space 68
3.3.2 Circular relationship between eight basic patterns for simple critical points 70
3.3.3 Simple critical points in α-β space 80
3.3.4 Regular points in α-β space 82
3.3.5 Multiple critical points in α-β space 82
3.3.6 The Growth of a Vector Field 82
3.3.7 The Interference between Critical Points 86
3.4 Earth Mover's Distance 86
3.5 Display of EMDs for a Large Set of Vector Fields 92
3.6 Applications 92
3.6.1 Unsteady Flow Past a 2D Cylinder 97
3.7 Chapter Summary 97

4 Topology of 3D Tensor Fields 103
4.1 Decomposition of a tensor field 104

4.2 Physical meaning of a deviator and an isotropic tensor 109
4.2.1 Isotropic Tensor 109
4.2.2 Deviator 110
4.2.3 Stress Tensor *vs.* Viscous Stress Tensor—An Example 110
4.3 Degenerate Points in a Deviator 113
4.4 The Existence Conditions of Hyperstreamlines in Q-R Space 115
4.5 Control functions of the singular points 119
4.6 The nature of a degenerate point 120
4.7 Representation of singularities in deviators 121
4.8 Topological Structure of a Singular Point in a 3D deviator 124
4.8.1 The Separating Surface of a Singular Point 124
4.8.2 Topological Skeletons of a Singular Point 126
4.9 Applications 133
4.10 Chapter Summary 141

5 Conclusions and Future Research **143**
5.1 Contributions 143
5.1.1 Feature Comparisons for Vector Fields 143
5.1.2 Decomposition of a Tensor Field 145
5.1.3 Topological Structure of 3D Tensor Fields 145
5.1.4 Representations of Degenerate Points 146
5.2 Analysis Framework 146
5.3 Future Research 149

A Related Publications **152**

Bibliography **154**

List of Tables

2.1 Rotation of the eigenvectors during a counterclockwise motion around a degenerate point. 51

3.1 Locations of three critical points and their corresponding α, β values; Order of critical point: the order of the lowest non-zero terms in the Taylor expansion of a critical point. 91

3.2 Locations of four critical points and their corresponding α, β values; Order of critical point: the order of the lowest non-zero terms in the Taylor expansion of a critical point. 91

List of Figures

1.1 Arrows depicting the velocity field in the flow past a cylinder. Blue, green, yellow, and red correspond to increasing energy density 15

1.2 Streamlines of the steady flow in Figure 1.1. Color maps kinetic energy density. 16

1.3 Streamtube colored with air pressure, visualizing the flow in a room (top); comparison of a streamline, a streamribbon, and a streamtube (bottom). 18

1.4 Streamlines of the steady flow in Figure 1.1. Color maps kinetic energy density. 19

1.5 Streamsurface in the flow around a post. 20

1.6 Field of nine individual components of a deformation tensor 22

1.7 Contour lines of the components of the stress-tensor field induced by two compressive forces. 23

1.8 Contour lines of the tensor field in Figure 1.7, after a 45° rotation of the coordinate axes. 24

1.9 Ellipses with principal axes representing the two eigenvectors $\vec{v}_i$ of an elastic stress-tensor field. 25

1.10 2-D and 3-D symmetric tensor fields. Orthogonal eigenvector fields $\vec{v}_i$ are represented as bidirectional arrows. 26

1.11 An example of a hyperstream line. λ_1 is in the direction of $\vec{v}_l$; λ_2 and λ_3 are $\vec{v}_{t_1}$ and $\vec{v}_{t_2}$, respectively 27

2.1 An example of a 2D vector field 31

2.2 Winding number of a vector field on a close path L 33

2.3 Some examples of Poincaré indices of different types of critical points. 34

2.4 Three basic sectors 35

2.5 A typical isolated critical point 35

2.6 2-D critical points. R1 and R2 denote the real parts of the eigenvalues of **J**, I1 and I2 the imaginary parts. 37

2.7 3-D saddle/saddle/node. 38

2.8 Topological representation of the 2-D flow past a circular cylinder at two different time steps. The flow is coming from the left; at = attachment point; de = detachment point; sp = saddle point; ce = center. . 40

2.9 Topological surfaces depicting the time evolution of the 2-D flow past a circular cylinder. Time increases from back to front. 41

2.10 Skin-friction topology on the upper hemisphere of the cylinder in Figure 1.1. Sa, Ss = saddles of attachment and separation; Na, Ns = nodes of attachment and separation; Fa, Fs = spiral nodes of attachment and separation. 42

2.11 Streamsurfaces depicting separation topology. 43

2.12 Away from degenerate points, continuous and symmetric tensor fields are diffeomorphic to constant data. 47

2.13 The two orthogonal eigenvector fields $\vec{v}_i$, represented as bidirectional arrows. 48

2.14 Hyperbolic (α_i), parabolic (β_j), and elliptic (γ_k) sectors at a degenerate point. 50

2.15 Simple degenerate points. $\delta = ad - bc \neq 0$ and $I_{\mathbf{T}}$ = tensor index. Trisector ($\delta < 0$) and wedge ($\delta > 0$) points. Trajectories s_k are separatrices. 53

2.16 Three-dimensional wedge point and trisector around a degenerate point where $\lambda_1 = \lambda_2 > \lambda_3$. 55

2.17 Frames of motion of degenerate points of the stress-tensor field over time. Wedge points are represented as black dots and trisectors are white dots. Color encodes the most compressive eigenvalue, λ_2, from very compressive (red), to mildly compressive (orange, yellow, green), to little compressive (blue). 59

2.18 An instantaneous topology of the most compressive eigenvector $\vec{v}_2$. Color encodes λ_2, from very compressive (red), to mildly compressive (orange, yellow, green), to little compressive (blue). W = wedge points, T = trisectors. 60

3.1 8 basic patterns for simple critical points 69

3.2 From (a)-(e), μ decreases from positive (a), (b) to zero (c) to negative (d), (e) 75

3.3 The point (α, β) and its equivalent angle Θ in $\alpha - \beta$ space. 78

3.4 Basic patterns for critical points in α-β space; *C* for center, *RN* for node, *AN* for attracting node, *RF* for repelling focus, *AF* for attracting focus, *St* for star, *Sa* for Saddle and *R* for regular point 80

3.5 A dipole point. 83

3.6 (a) an attracting focus (b) a saddle point (c) a complex field created by adding (a) and (b) to a constant field. 85

3.7 A vector field with three simple critical points. 89

3.8 A vector field with four critical points and five corresponding points in α-β space. 90
3.9 eight basic patterns of simple critical points sorted by their EMDs . . 93
3.10 sixteen different vector fields 95
3.11 sixteen different vector fields sorted by MDS 96
3.12 The EMDs for flows in different time steps with time step 3. The EMD for time step 3 and 18 is zero, which means that flow at these two time steps have the same topology. 98
3.13 An incompressible viscous flow past a 2D circular cylinder. 99
3.14 The flow at time step 3 (top) and time step 18 (bottom). 100
3.15 The flow at time step 24 (top) and time step 18 (bottom). 101

4.1 Eigenvector e_3 is aligned to x_3, a local 2D tensor on the plane. 106
4.2 Stress tensor induced by two compressive forces; minor hyperstreamlines 108
4.3 Stress field in a flow past a hemisphere cylinder. 111
4.4 Stress field in a flow past a hemisphere cylinder. 112
4.5 (a) $A(\lambda)$ is a general cubic polynomial with three real distinct roots; (b) $A(\lambda)$ has one double root and one single root: $\lambda_1 = \lambda_2 > \lambda_3$ (c) $A(\lambda)$ has one double root and one single root: $\lambda_1 > \lambda_2 = \lambda_3$; (d) $A(\lambda)$ has one triple root: $\lambda_1 = \lambda_2 = \lambda_3$. 116
4.6 Q'-R' plane 118
4.7 Representation of triple and double degeneracy. (top) a line and a cone for $\lambda_1 = \lambda_2 > \lambda_3$; (bottom) a line and a cone for $\lambda_1 > \lambda_2 = \lambda_3$. 125
4.8 A point of triple degeneracy with 6 hyperbolic separating surfaces. . . 127
4.9 A point of triple degeneracy with 2 hyperbolic separating surfaces and one parabolic surface. 127
4.10 Stress tensor induced by two compressive forces; minor hyperstreamlines 128

4.11 Stress tensor induced by two compressive forces; major hyperstreamlines 130
4.12 Stress tensor induced by two compressive forces; minor hyperstreamlines 131
4.13 Stress tensor induced by two compressive forces; medium hyperstreamlines 132
4.14 A flow past a wingtip: Major eigenvector field (top); Minor eigenvector field (bottom). 135
4.15 Deformation tensor in a flow past a hemisphere cylinder at incidence 136
4.16 Deformation tensor in the near field of a flow past a wingtip; major eigenvector field (top) and minor eigenvector field (bottom) 138
4.17 Reynolds stress tensor in the near field of a flow past a wingtip; major eigenvector field (top) and minor eigenvector field (bottom) 139

5.1 A frame work for feature comparisons of vector fields. 147
5.2 A frame work for tensor field analysis. 148

Chapter 1

Introduction

1.1 Scientific Visualization

"The purpose of computing is insight, not numbers" [1]

[Richard Hamming, 1962]

Scientific visualization transforms complex data obtained from numerical simulations, observations and experiments into a graphical abstraction and renders it into an image amenable to understanding by the human visual system while maintaining the integrity of the information.

The concept of visualizing numerical data did not appear recently. From the middle of 17th century to the beginning of 20th century, there have been techniques developed by some of the greatest scientists including Halley, Watt, Descartes, Lambert, Playfair and von Humboldt. However, it was not until the introduction of computer graphics in the 1960s, when data visualization was brought into the forefront of science and engineering. In particular, McCormick's report on *Visualization in Scientific Computing* [2] in 1987 stimulated an explosion of research in this "new" field. Today, computer generated images have become a standard for communication

involving human perception and abstract representation of the physical world.

Multidimensional Multivariate Visualization

Multidimensional multivariate (mdmv) visualization is an important subfield of scientific visualization. It was studied separately by statisticians and psychologists long before computer science was deemed a discipline [3]. This subfield studies multiple parameters and the key relationship between them. *Multidimensional* refers to the dimensionality of the independent variables, and *multivariate* refers to the dimensionality of the dependent variables [4]. For example, a deformation tensor field (symmetric tensor) observed and recorded in a three-dimensional space at various locations produces $3d6v$ data.

Scientists have studied multivariate visualization since 1782 when Crome used point symbols to show the geographical distribution in Europe of 56 commodities [5]. However, before Tukey's exploratory data analysis, tools for multivariate visualization usually consisted of colored pencils and graph paper, and mainly dealt with small-sized one or two variate data. The appearance of low-priced personal computers and workstations during the 1980s breathed new life into graphical analysis of mdmv data. During the last decade, hundreds of *new* mdmv visualization techniques have been invented [6]. Among them, a large number have been devoted to vector and tensor field visualization such as described in [7, 8, 9, 10, 11, 12, 13, 14, 15].

1.2 Vector and Second-order Tensor Fields

Vectors and second-order tensors are both multidimensional, multivariate quantities. A vector field in a three-dimensional space has three components at each location in the field, therefore generates $3d3v$ data; whereas a general tensor field in a three-dimensional space has nine components and generates $3d9v$ data. In some special

cases, the number of independent components of a tensor is reduced. For example, a symmetric tensor has only six independent components and an anti-symmetric tensor has only three, therefore produce $3d6v$ and $3d3v$ data, respectively.

1.2.1 Vectors, Second-order Tensors and the Field Concept

A tensor is a quantity whose key property is the *transformation* law of its components, i.e, the way its components in one coordinate system are related to its components in another. The order of a tensor describes its complexity [16]. The simplest tensors are a scalar and a vector which are a zeroth-order and a first-order tensor, respectively. A second-order tensor is next in order of complexity in the tensor family. The focus of this dissertation will be on vectors and second-order tensors. [1]

Vector

A vector is a quantity uniquely specified in any coordinate system by three real numbers (or three components). The components of a vector transform under changes of the coordinate system according to the law (Einstein summation rule is used here):

$$V_i' = \alpha_{ik} V_k$$

where V_i', V_k are the components of the vector in the old and new coordinate systems K and K', respectively, and α_{ik} is the cosine of the angle between the ith axis of K' and the kth axis of K. Generally, if a vector vanishes in one coordinate system it vanishes in other coordinate systems as well [16].

[1] in this dissertation, "second-order tensor" will be very often referred as just "tensor" for convenience.

Second-order Tensor

A second-order tensor is a quantity uniquely specified by nine real numbers (the components of the tensor) which transform under changes of the coordinate system according to the law:

$$T'_{ik} = \alpha_{il}\alpha_{km}T_{lm} \tag{1.1}$$

where T_{lm} and T'_{ik} are the components of the tensor in the old and new coordinate systems K and K', respectively, and α_{il} is the cosine of the angle between the ith axis of K' and lth axis of K (similarly for α_{km}) [16]. Generally, if all the components of a tensor vanish in one coordinate, they also vanish in any other coordinate systems.

A second-order tensor is often expressed as a matrix:

$$\begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} \tag{1.2}$$

Tensor Field Concept

A tensor field is a rule assigning a unique value of a tensor to each point of a certain volume V in space. Let **r** be the position vector of a variable point of V with respect to the origin of some coordinate system. Then a tensor field of order n is indicated by

$$T_{i_1 i_2 \ldots i_n} = T_{i_1 i_2 \ldots i_n}(\mathbf{r}) \tag{1.3}$$

Thus, a vector field is defined as:

$$T_{i_1} = V_{i_1}(\mathbf{r}) \tag{1.4}$$

And a second-order tensor field is defined as:

$$T_{i_1 i_2} = \begin{pmatrix} T_{11}(x,y,z) & T_{12}(x,y,z) & T_{13}(x,y,z) \\ T_{21}(x,y,z) & T_{22}(x,y,z) & T_{23}(x,y,z) \\ T_{31}(x,y,z) & T_{32}(x,y,z) & T_{33}(x,y,z) \end{pmatrix} \tag{1.5}$$

Definition 1 (Critical Points) *A critical point is a point in a vector field where all three vector components are zero and the streamline slope is indeterminate.*

Critical points are the only points in a vector field where tangent curves may cross each other. In the case of a steady state velocity field, tangent curves represent streamlines. Critical points are characterized according to the behavior of nearby tangent curves. There are eight basic patterns for simple (first order) critical points[1], namely, an attracting/repelling star, an attracting/repelling node, a saddle, an attracting/repelling focus and a center. A particular set of tangent curves-separatrices[1], which end on critical points, are of special interest because they define the skeleton which characterizes the global behavior of all other tangent curves in the vector field. The number and nature of critical points in a vector field remain unchanged under a continuous transformation.

Definition 2 (Degenerate Points) *A degenerate point of a tensor field* $\boldsymbol{T} : E \rightarrow \mathcal{L}(\boldsymbol{R}^m, \boldsymbol{R}^m)$, *where* E *is an open subset of* $\boldsymbol{R}^m$, *is a point* $\vec{x}_0 \in E$ *where at least two of the* m *eigenvalues of* $\boldsymbol{T}$ *are equal to each other [38].*

In the case of two-dimensional tensor fields, there are only two eigenvalues λ_1 and λ_2, and $\vec{x}_0$ is a degenerate point iff $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0)$. For three-dimensional tensor fields, various types of degenerate points exist, corresponding to the conditions $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0)$, $\lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0)$, or $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0)$.

[1] will be explained in Chapter 2

1.2.2 Combinatorial Topology of Vector and Tensor Fields

Combinatorial topology, also known as "rubber sheet geometry", is a branch of geometry. It studies the properties of figures that endure when the figures are subjected to continuous transformations. A topological property of a figure is a property possessed alike by the figure and all its topological equivalents. In combinatorial topology, complicated figures are constructed from simple ones and their properties are deduced from the simple figures [17].

Combinatorial topology has extensive applications in geometry and analysis, many of which result from connections with the theory of differential equations. It may seem surprising that such superficially different subjects as topology and differential equations could be related, but research has shown that a link between these two subjects is the concept of a vector field [17]. Recent developments in scientific visualization have shown that vector fields and their topological structures also play a very important role in analyzing second-order tensor fields.

Second-order tensors are fully represented by their eigenvectors and associated eigenvalues.

$$\mathbf{T}\vec{e_i} = \lambda_i \vec{e_i} \tag{1.6}$$

where λ_i and $\vec{e_i}$ $(i = 1, 2, 3)$ are eigenvalues and eigenvectors of the tensor $\mathbf{T}$, respectively. The λ_i's represent all the amplitude information while the $\vec{e_i}$'s represent all the directional information of $\mathbf{T}$. Visualizing a tensor field is equivalent to visualizing its eigenvector fields. However, unlike a vector field, eigenvectors are vectors with sign indeterminacy. This remarkable feature distinguishes eigenvector fields from ordinary vector fields and makes their topological features even simpler. The basic constituents of tensor topology are degenerate points where at least two of the eigenvalues are equal to each other. They play a role similar to critical points in vector fields.

1.3 Motivations and Objectives

1.3.1 Overview of vector field visualization

Vector fields have vast applications in physical sciences. The force fields arising from gravitation and electromagnetism; the velocity vectors of a fluid motion, found for example over an airfoil; and gradients, such as the pressure gradient appearing on a weather map are all elements of vector fields.

Vector field visualization has received an enormous amount of attention. Traditionally, visualization has been part of measurements in complex physical experiments. Experimental fluid mechanics relies heavily on physical visualization for flow measurements and analysis. Many techniques have been developed in the past: seeding a flow with smoke or dye dates back decades [18]; using hundreds of tiny tufts attached to screens for visualization of two-dimensional cross sections of flows [19]; and photographs of oil streak patterns on body surfaces. These techniques provide a wealth of information for both local and global flow structures and have had a profound influence on flow modeling studies. However, direct results from experimental visualization usually give us only a qualitative measure and the derivation of quantitative information is accomplished by image processing and analysis. And more recently, with the fast development of computer graphics, data visualization and analysis have provided both qualitative and quantitative insight about fluid flows [20, 21]. Many effective ways of conveying large vector field data into geometric objects, such as arrows, glyphs, hedgehogs, streamlines, streamsurfaces and texture mapping, have been developed over the past two decades. In recent years, critical point theory based topological methods have been introduced into vector field visualization for better understanding of the local and global structures of vector fields. Flow visualization also serves as the basis for classification of fluid flows [22, 8, 9, 23, 10, 24, 25].

Despite all the great developments in vector field visualization, to our knowledge,

there hasn't been any research on quantitative measure of closeness between vector fields. The usefulness of this measurement can be seen when comparing computational and experimental flow fields under the same conditions for design purposes or for the verification of theories. Furthermore, its applicability can be extended to more cumbersome tasks such as navigating through a large database while searching for similar topology. It might also be used to better quantify the changes in time varying data sets.

1.3.2 Overview of tensor field visualization

As with vector fields, there are numerous second-order tensor quantities that are of interest for analysis and visualization. In fluid mechanics, the Reynolds stress and the strain-rate (deformation) tensors are tensors of considerable interest in turbulence studies. The alignment of an medium eigenvector[2] of a deformation tensor with the vorticity vector and scalar gradients [26] [27] provides a way of vortex detection. In solid mechanics, the study of the Boussinesq tensor tells us the force distribution inside a solid body. And in general relativity, the basic field equations are described by the Einstein tensor and energy-momentum tensor.

Second-order tensor data sets are at the heart of many engineering and physical disciplines. Yet very few methods have been developed to understand and visualize such data sets due to their complex nature and their large size. In traditional approaches, data are displayed using simple local icons depicting, for example, eigenvectors and eigenvalues at discrete locations. Alternatively, scalar components are analyzed and displayed in two- or three-dimensional form as an aid to understanding these data. These classical methods are far from effective, but are almost universally used to try to extract important information about underlying physical processes and engineering principles. More significantly, through this data presentation process,

[2]see sec:hyper

valuable information such as continuity of the data and global structure is discarded. These difficulties provided us with a substantial motivation to find efficient methods for analyzing and displaying them. Thierry Dermarcelle introduced topological methods for analyzing and visualizing second-order tensor data [11]. This approach preserves the continuity of the tensor field and reveals the key properties of the field without any information loss. A new visual representation-hyperstreamline [12] has also been designed in order to visualize the topological skeletons of the tensor field. However, his topological study is mainly applicable to two-dimensional space and concentrates only on simple, isolated degenerate point theory in the field. We live in a three-dimensional world and there are also a lot of cases where degenerate points are continuous. Therefore, it is important for us to understand the underlying physics of this world.

1.3.3 Objectives

Our main objectives in this research are as follows:

- Develop a quantitative measure for feature comparisons of vector fields,
- Design a good preprocess procedure to simplify the tensor analysis,
- Study the topological structure of three-dimensional tensor fields, and
- Find the representations of continuous degenerate points.

Vector Field Comparisons

The concept of critical points in vector fields has been well studied and has proven to play a key role in vector field topology. However, there hasn't been any study on quantitative measurement of similarities and differences between vector fields using topology. In fluid dynamics, simplifications are necessary in the numerical simulation

process in order to compute analytical solutions of flows over complex bodies in a reasonable amount of time. Researchers formulate various models with the hope of capturing the essential features of real flows. For example, Reynolds stress and velocity fields in turbulent flows are almost impossible to solve analytically. Many turbulence models, both linear and nonlinear, have been suggested to better capture real flows.

For example, in eddy-viscosity models, the average velocity gradient is related to velocity

$$-u_i\bar{u}_j = \nu_t(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) - \frac{2}{3}k\delta_{ij}, \quad i,j = 1,2,3 \tag{1.7}$$

where ν_t=(turbulent velocity scale) $\times$ (turbulent length scale) and k is the eddy-viscosity. This approach to turbulence closure is extremely attractive from a computational point of view, especially in terms of numerical robustness. Second-moment closure models (originated by Pope [28]) have been pursued extensively over the past decade, with recent efforts being exemplified by the models of Speziale et al' [29], Fu et al' [30], Launder and Tselepidakis [31], Craft and Launder [32] and Hanjalic and Jakirlic [33]. All these models are derived from the exact Reynolds-stress-transport equation and are mathematically elaborate but computationally expensive. Recent developments in the construction of nonlinear eddy-viscosity formulation, such as *The Lien-Leschziner Model* [34], *The Craft-Suga-Launder Models* [35, 36] and *The Apsley-Leschziner Model* [37], have been made in order to combine the simplicity of the eddy-viscosity formulation with the superior fundamental strength and predictive properties of second-moment closure.

However, all these models for a large part have been quite arbitrary. In order to test the validity of a certain model, a comparison between the computational and experimental results is necessary. However, a metric for the closeness between the computational and experimental results remains unsolved. Traditionally, a series of

charts for each component of Reynolds stress and the velocity vector have been drawn. It is perceptually nonintuitive, as well as difficult to show the intrinsic structural differences between the two fields. This dissertation fills the void of not having a measurement for feature comparison between vector fields and for second-order tensor fields.

Decomposition of Tensor Fields As a Preprocessor

Tensor analysis is a very important yet underdeveloped area in visualization. The sheer enormous volume of information and the multivariate nature have been the major hindrances. How to simplify the analysis and extract a small fraction of the data sets without loss of information becomes the key issue. Every tensor field can be decomposed into a deviator and a spherical part (definitions to follow). The spherical part is an isotropic tensor and therefore remains invariant to a coordinate system transformation. As a result, there is no particular need to study this part of the tensor (here we refer to it as "the isotropic tensor"), it serves as a uniform bias. It will be shown that the deviator of a tensor is parallel to the tensor itself. Therefore, their respective eigenvector fields are identical. Furthermore, the locations of the respective degenerate points are also identical. This, in turn, means that the topology of a tensor field is identical to the topology of its deviator. A deviator determines the properties of a tensor field.

In this dissertation, extraction of a deviator from a tensor field is described as a preprocessor for tensor analysis.

Topology of Three-Dimensional Tensor Fields

The difficulty with representing tensor data arises from their complexity. At a point in space one frequently has to deal with nine or more components of a physical variable or a derived quantity such as vorticity, or stress; and there are usually millions of

data points available from computation or experiment. In the past, therefore, with few techniques available for tensor data analysis and display (often only on a local basis), very little information was obtained from complex calculations concerning fluid flows, electromagnetic radiation, or field quantities in general relativity. Much more effort is usually expanded on the generation of data than on the understanding that can be attained from them. In certain fields of physics, this problem is particularly acute. For example, in non-Newtonian fluid mechanics (such as the flow of plastics) the constituent equations are often not known, and the tensor data describing stress and strain are dependent on the model used for these equations. It would be very significant to have methods that can analyze the results of these calculations while facilitating insight into the underlying physical phenomena; for example, by efficiently analyzing and displaying tensor data sets it may be possible to investigate the effect of various forms of the constituent equations on the resulting stress and strain tensors in non-Newtonian flows.

Topological representations have proven to be very useful in fluid mechanics as well as in engineering, physics and mathematics. The usefulness is derived from the simplicity of the representation, while preserving the richness of the data. A skillful designer can make very good use of topological skeletons by having an intuitive understanding as to how these skeletons relate to physical flow features which give rise to large scale effects such as flow separation, or drag and lift on lifting bodies.

Tensor field topology has not been studied in much detail for the purpose of visualization. Therefore we have devised methods, based on mathematical foundations, that allow the topology of symmetric tensor fields to be determined. In two dimensions, this process is straight forward, and reveals simple geometric pictures that represent tensor data locally as well as globally in a very simple manner, usually on a surface [11, 12, 13]. In three dimensions, this process is significantly more difficult to carry out and theoretical notions are not easily derived. Yet, the results are

strikingly simple and effective in those cases that we have studied so far. In fact, tensor field topology is often simpler than vector field topology, a counter-intuitive result given that tensor data appear more complex and rich than vector data sets. Three-dimensional tensor field topology will be discussed in detail in Chapter 4.

Representation of Continuous Degenerate Points

Previous studies of tensor field topology only deal with isolated degenerate points in two-dimensional case [38]. However, points of double degeneracy in three-dimensional space may appear along lines or surfaces. In this dissertation, the concept of a "control function" of a *deviator* is proposed. This function determines the existence of degenerate points and assists in defining the lines and surfaces along which the degenerate points lie. Following the study of the tensor field in the neighborhood of degenerate points, one can display a full representation of the three-dimensional tensor field.

1.4 Visualization Techniques for Vector and Tensor Fields

In this section, we discuss a few new developments in visualization techniques and show their advantages over the conventional methods. For vector field visualization, the most straight forward method is drawing point icons; while for second-order tensor fields, the traditional method is drawing contour plots for individual components or glyphs for eigenvectors at discrete locations. All these techniques have their limitations for visualizing continuous multivariate data sets. Streamlines (including streamtubes and streamsurfaces) and hyperstreamlines are new visual representations for vector and tensor fields, respectively. These techniques are designed especially for

displaying multidimensional multivariate information while maintaining continuity. Texture mapping is a very good way of rendering two-dimensional or $2D$ slices of three-dimensional continuous data sets, both for vector and tensor fields.

1.4.1 Vector Fields

Point Icons

Point icons are useful in visualizing 2D slices of 3D vector fields. There are several candidates such as arrows, wedges and hedgehogs. A comparative study shows that arrows are most efficient at conveying vector information from volumetric data sets [39]. This method is simple and straight forward, but is impractical when applied to the entire volume because of the visual clutter. Therefore, the density of displayed arrows must be kept low. However, this make it difficult to comprehend the underlying structure of the vector field by mentally interpolating adjacent arrows.

Figure 1.1 shows the velocity field of the flow past a circular cylinder. Arrows represent the velocity vector at various locations. Color maps kinetic energy density. Although valuable information is depicted, the display of merely 4% of the arrows renders the image overly cluttered and does not convey the intrinsic continuity of the data.

Streamlines, Streamtubes and Streamsurfaces

Line icons are more efficient in the sense that they provide a continuous representation of the data. Consider a vector field $\vec{v}(\vec{x}, t)$ at time t_0, where streamlines are integral curves satisfying:

$$\frac{d\vec{x}}{ds} = \vec{v}(\vec{x}, t_0) \tag{1.8}$$

where $\vec{x}$ is the position in space and s is a parameter measuring distance along the path. Streamlines are everywhere tangent to the flow $\vec{v}(\vec{x}, t_0)$.

Multiple Pages Missing from Available Version

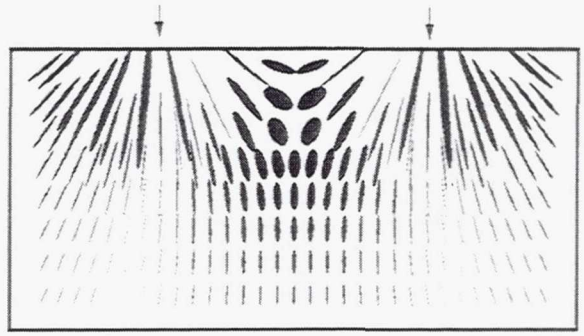


Figure 1.9: Ellipses with principal axes representing the two eigenvectors $\vec{v}_i$ of an elastic stress-tensor field.

of the axes of the ellipses/ellipsoids represent the two/three-dimensional eigenvector fields (eigenvectors and eigenvalues) $\vec{v}_i$ (Figure 1.9) of the tensor field. The advantage of this representation is that it remains invariant under rotation. However, ellipses/ellipsoids suffer from the the same limitation, namely, visual clutter (Figure 1.9, as arrows in vector fields.

Hyperstreamlines

The topology of a tensor field $\mathbf{T}(\vec{x})$ is the topology of its eigenvector fields $\vec{v}_i(\vec{x})$, $i = 1, 2, 3$ [16]. Line icons representing eigenvector fields were introduced by Thierry Delmarcelle [11, 12] and are able to embed the multivariate information of tensor fields along trajectories in $3D$ space.

Definition 3 (Hyperstreamline) *A geometric primitive of finite size sweeps along the longitudinal eigenvector field, $\vec{v}_l$, while stretching in the transverse plane under the combined action of the two transverse eigenvectors, $\vec{v}_{t_1}$ and $\vec{v}_{t_2}$. Hyperstreamlines are surfaces that envelop the stretched primitives along the trajectories. We color hyperstreamlines by means of a user-defined function of the three eigenvalues, usually the amplitude of the longitudinal eigenvalue. [38]*

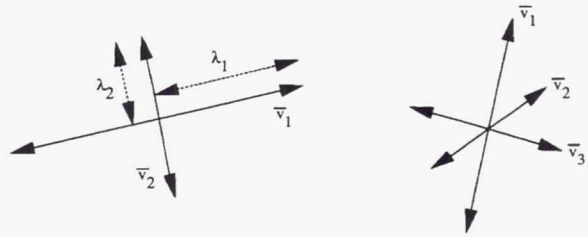


Figure 1.10: 2-D and 3-D symmetric tensor fields. Orthogonal eigenvector fields $\vec{v}_i$ are represented as bidirectional arrows.

Color and trajectory of hyperstreamlines represent the longitudinal eigenvector field and cross-sections encode the transverse eigenvectors (Figure 1.11). Thus, hyperstreamlines fully represent tensor data along continuous trajectories. Similar to streamlines, hyperstreamlines are generated by integrating the eigenvector fields. Hyperstreamlines are referred as "major," "medium," or "minor," depending on the corresponding longitudinal eigenvector field that defines their trajectories.

1.4.3 Texture Mappings

Textures have been well known in flow visualization. The simulation of particle convection leads to texture [43, 44, 45]. The clouds, smoke and other typical textures are perceived in experimental visualization when many particles are used. However, texture mapping is a recently developed technique in the area of scientific visualization. It renders streamlines/hyperstreamlines without integrating the vector/eigenvector field and gives a continuous, space-filling view of a $2D$ field or $2D$ slices in a $3D$ field. The drawback of texture mapping is that it's viewpoint dependent.

Many texture mapping techniques have been proposed. Van Wijk's *spot noise*

Multiple Pages Missing from Available Version

Chapter 4 discusses the topology of $3D$ second-order tensor fields. First, a decomposition procedure is introduced. It serves as a preprocessor to simplify the tenor analysis. Second, the existence conditions of trajectories are discussed in $Q-R$ plane. Third, the representations of continuous degenerate points in $3D$ are defined by control functions. Finally, the basic structures about simple degenerate points in $3D$ space are discussed.

Vector and tensor field visualization is a relative new yet very important area. Visual representations are most efficient ways for scientists and engineers understanding complex physical phenomena. Chapter 5 summarizes the contributions of the current research and discusses the future research issues.

Chapter 2

Review of Vector and Tensor Field Topology

Topology is the basis for the visualization of vector and second-order tensor fields. The purpose of visualization is to help scientists and engineers understand the behavior of trajectories in a certain domain. Topology methods seek the surfaces in $3D$ or the curves in $2D$ which divide the domain into regions where trajectories behave differently. The set of dividing curves or surfaces defines the vector/second-order tensor field and is determined by the critical/degenerate points in the field.

This chapter is intended to give an overview of the state of the art of the research on vector and second-order tensor field topology. It is divided into two parts. The first part discusses the critical point theory and vector field topology in both $2D$ and $3D$ space; the second part discusses degenerate point theory and research on the development of tensor field topology.

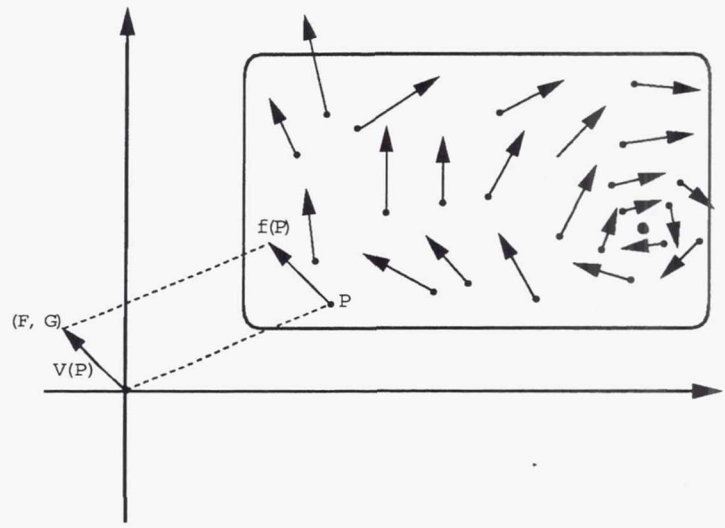


Figure 2.1: An example of a 2D vector field

2.1 Vector Field Topology

A **vector field** V on a subset D is a function assigning to each point P of D a vector with its tail at P [17]. Figure 2.1 shows an example in a $2D$ plane. Placement of the vector V(P) with its tail at P is mainly a dramatic visual aid, which adds little quantitative information about the field. The essential qualities of the vector V(P) are its length and direction. Place all vectors with their tails at the origin, then V(P) is described as:

$$V(P) = (F(x, y), G(x, y)) \tag{2.1}$$

where F and G are real-valued functions of $P = (x, y)$.

2.1.1 Critical Point Theory

A vector field 2.1 determines a system of differential equations:

$$\begin{cases} \frac{dx}{dt} &= F(x,y) \\ \frac{dy}{dt} &= G(x,y) \end{cases} \tag{2.2}$$

in some region D. The solutions form a family of *directed paths*, called **integral paths** of the system, which are tangent to the vector field at each point P. There is exactly one integral path that passes through each point P at which V(P) is not zero. The picture formed by these paths is called the **phase portrait** of the system of differential equations. The **phase portrait** is determined by particular points P, called **critical points**[1], where $V(P) = 0$ and around which the integral paths gather [17].

The most important characteristics of the phase portrait are the number and arrangement of the critical points, the pattern of the integral paths about each point, and the stability of the critical points. The number and nature of critical points won't change under continuous transformations. These are topological properties of the system of differential equations. To characterize the topology, a number of definitions and theorems are introduced in the following sections.

Winding Number

Consider a continuous vector field V on a closed path L with no zero on L. Starting at a fixed point P on L, the vector V(P) will wiggle about during the trip around L and return to the point P with some **whole** number of revolutions (Figure 2.2).

Proof: The vector V(P) starts at P and after traversing the path L, it returns to the same position and with the same direction and length as the original V(P). Therefore, the number of revolutions during the trip has to be integers.

This completes the proof. ◇

[1]see Chapter 1, section 1.2.1

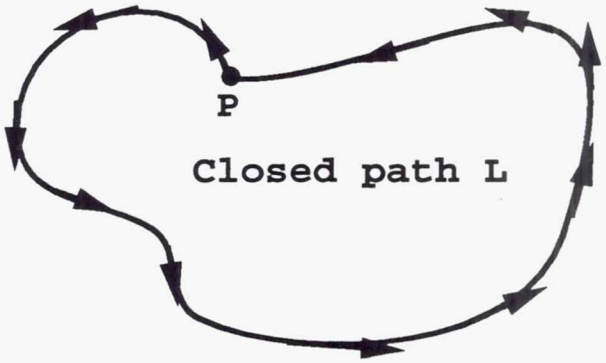


Figure 2.2: Winding number of a vector field on a close path L

Counting these revolutions positively if they are counterclockwise or negatively if they are clockwise, the resulting algebraic sum of the number of revolutions is called the **winding number of V on *L***, denoted as W(L) [17].

Isolated Critical Points and Poincaré Index

Definition 4 (Isolated Critical Points) *If for a critical point P, there exists a neighborhood of that vector field V in which the vector field vanishes only at P, then P is an isolated critical point.*

The Poincaré index of V at P, denoted I(P), is defined as the winding number W(γ) of V on γ.

Figure 2.3 shows some examples of Poincaré indices of different types of critical points.

Theorem 1 (The Poincaré Index Theorem) *Let V be a continuous vector field. Let D be a cell and γ its boundary. Supposing that V is not zero on γ, then*

$$W(\gamma) = I(P_1) + I(P_2) + \cdots + I(P_n) \tag{2.3}$$

where $P_1, P_2, ..., P_n$ are the critical points of V inside D.

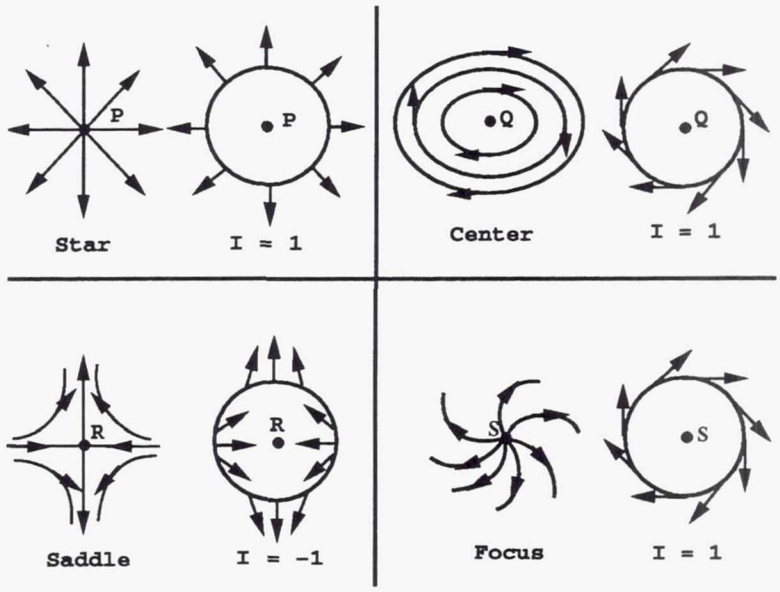


Figure 2.3: Some examples of Poincaré indices of different types of critical points.

The Poincaré index measures a property of the vector field that depends only on the behavior of V in an arbitrarily small area around P. It is only a crude means of classifying critical points, since points of widely different types may have the same index (Figure 2.3) [17].

Basic Building Blocks

All critical points are made up of sectors containing one or more of the three types shown in Figure 2.4: an **elliptic sector**, where all paths begin and end at the critical point; a **parabolic sector**, where just one end of the path is at the critical point; a **hyperbolic sector**, where the paths do not reach, just sweep past the critical point. These three sectors are the basic building blocks of the vector fields surrounding the critical points. The paths that divide each sector from the next are called **separatrices**.

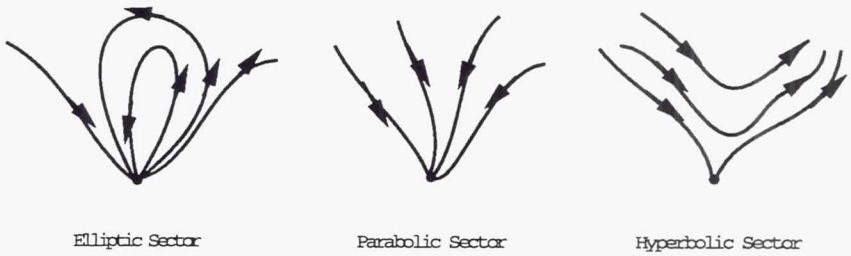


Figure 2.4: Three basic sectors

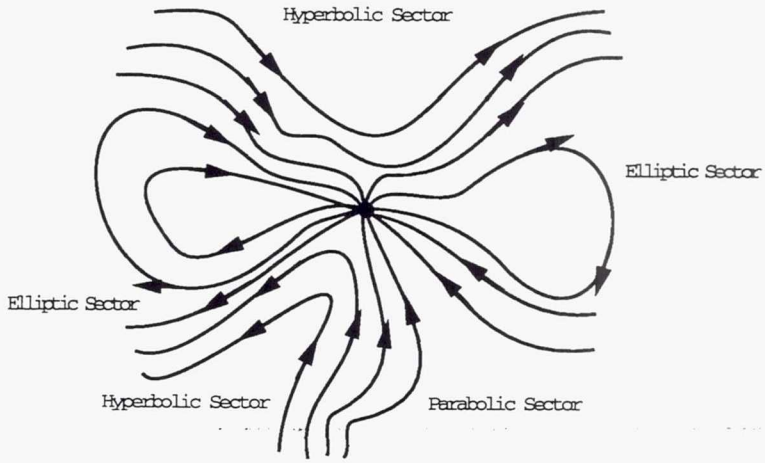


Figure 2.5: A typical isolated critical point

A typical critical point may have sectors of all three types (Figure 2.5). A critical point may have only one type of sector. Those with only parabolic sectors are called **nodes**. Those with only elliptic sectors are called **roses**, for example, a dipole. Those with only hyperbolic sectors are called **cross points**. Saddle points are cross points with four sectors. Still more complicated types are possible with an infinite number of sectors, and may be nonisolated critical points [17]. Here, only isolated critical points with a finite number of sectors are under consideration.

$2D$ Critical Points

Since a vector field vanishes at a critical point, the behavior of nearby streamlines is determined by the first order partial derivatives of the vector field. More precisely, the 2-D vector field $\vec{v} = (v_1, v_2)$ near a critical point is given, in most cases, by the first-order expansion

$$\begin{cases} v_1(dx_1, dx_2) \approx \frac{\partial v_1}{\partial x_1} dx_1 + \frac{\partial v_1}{\partial x_2} dx_2 \\ v_2(dx_1, dx_2) \approx \frac{\partial v_2}{\partial x_1} dx_1 + \frac{\partial v_2}{\partial x_2} dx_2 \end{cases}$$

where dx_1 and dx_2 are small distance increments from the critical point position. Thus, the nearby flow pattern is completely determined by the 2×2 Jacobian matrix $\mathbf{J}$, whose elements

$$J_{ij} = \frac{\partial v_i}{\partial x_j} \tag{2.4}$$

$(i, j = 1, 2)$, are evaluated at the critical point position.

Different patterns arise that are characterized by the eigenvalues of the matrix $\mathbf{J}$. Figure 2.6 shows how the eigenvalues of $\mathbf{J}$ classify a critical point as an attracting/repelling node, an attracting/repelling focus, a center, or a saddle. Real eigenvectors of $\mathbf{J}$ are tangent to the streamlines ending at the critical point. A positive or negative real part of an eigenvalue indicates an attracting or repelling nature, respectively. The imaginary part denotes circulation about the point.

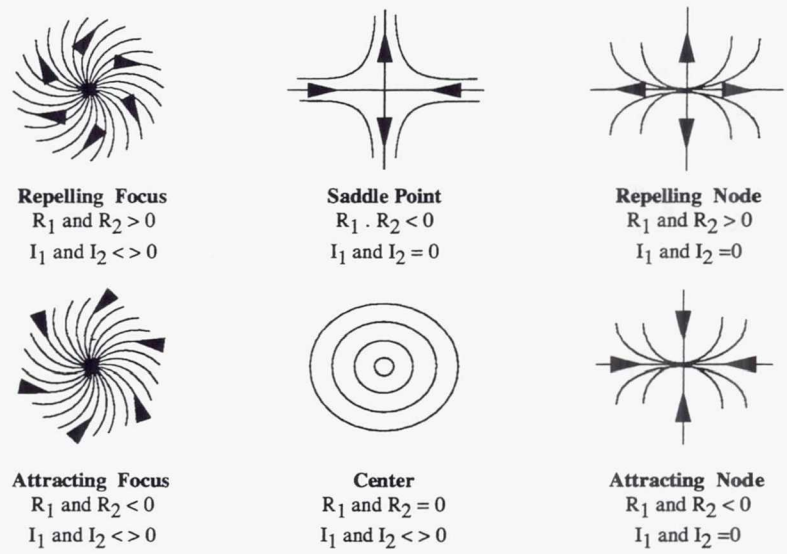


Figure 2.6: 2-D critical points. R1 and R2 denote the real parts of the eigenvalues of **J**, I1 and I2 the imaginary parts.

In $2D$ flows, critical points near the surface of a body known as "attachments" and "detachments" play a similar role as saddle points Figure 2.8. In $2D$ flows near the surface of a body where the velocity is constrained to be zero, the flow is mainly tangential and streamlines propagate parallel to the surface. The "detachment"/"attachment" appear at point where the tangential velocity vanishes and a streamline suddenly originates/terminates [53].

$3D$ Critical Points

Simple critical points of a $2D$ vector field give rise to a basic set of patterns shown in Figure 2.6. In $3D$, critical points can be classified by simple generalization [54] of the $2D$ classification shown in Figure 2.6. Alternatively, the classification can be done by examining the invariants of the matrix **J** [22].

For a $3D$ vector field defined over a $3D$ domain, $\vec{v} = (v_1, v_2, v_3)$, the Jacobian

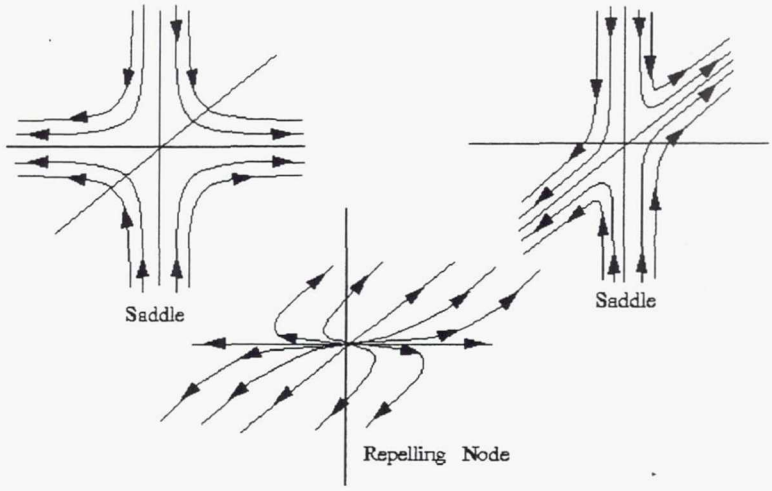


Figure 2.7: 3-D saddle/saddle/node.

matrix **J** is a 3×3 matrix whose elements are given by Equation 2.4, for $i, j = 1, 2, 3$. However, **J** has three eigenvalues and three eigenvectors. Complex eigenvalues always occur in conjugate pairs together with a third real eigenvalue. Again, real eigenvectors are tangent to streamlines ending at the critical point, and complex eigenvalues denote circulation.

Possible $3D$ patterns include purely attracting/repelling nodes with eigenvalues being all real and all negative/positive, and they behave as 2-D attracting/repelling nodes in each of the three planes spanned by pairs of eigenvectors; saddle-saddle-nodes with eigenvalues being all real but one having a different sign, and they behave as 2-D saddles in two planes and as a 2-D node in the third plane; and spiral nodes with one real and two complex conjugate eigenvalues, and have an attractive or repelling third direction. Figure 2.7 shows an example of an saddle-saddle-node. Algorithms for locating and extracting critical points can be found in [55, 56].

Unsteady Flow Fields. Critical points in unsteady vector fields move and eventually merge or split. These phenomena are studied by tracking and representing their trajectories over time [57], or by locating interactively critical points in nearby space-time regions [58].

2.1.2 Global Topology

Vector field topology depicts schematically the behavior of a large collection of streamlines. The basic constituents of vector field topology are

- the critical points (Section 2.1.1) and
- the set of their connecting streamlines (separatrices).

The topological skeletons divide the field into regions topologically equivalent to uniform flow. The representation is highly effective, due to the ease of inferring the behavior of every streamline in the flow from these simplified graphs. Also, comparing the flow at different time steps is greatly facilitated and can be automated using syntactic pattern recognition [20].

$2D$ Time-Dependent Fields

Streamsurfaces described in Section 1.4.1 can be used to visualize two-dimensional unsteady flows that depend on time, Reynolds number or any other parameter with the third dimension corresponding to the parameter value. Generated by stacking instantaneous topological skeletons, streamsurfaces are able to display the topological transitions that may occur between consecutive steps.

Figure 2.9 shows the surfaces in a simulated flow about a $2D$ cylinder. Time increases from back to front. The surfaces are obtained by connecting the streamlines and the third dimension corresponding to time. Yellow and blue surfaces correspond

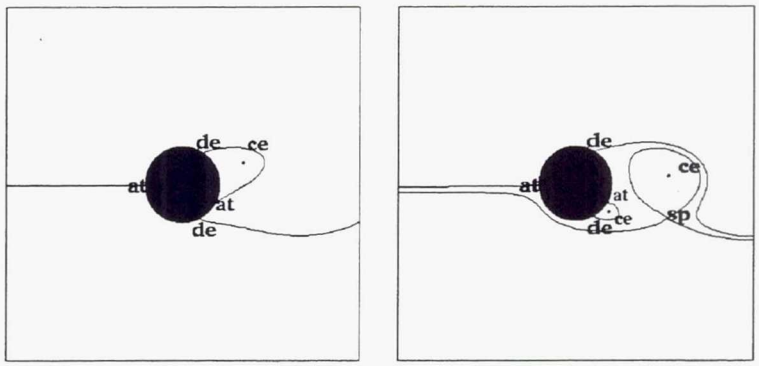


Figure 2.8: Topological representation of the 2-D flow past a circular cylinder at two different time steps. The flow is coming from the left; at = attachment point; de = detachment point; sp = saddle point; ce = center.

Image courtesy of J. Helman (Reference [10]).

to incoming and outgoing separatrices, respectively. Surfaces from attachment points are colored orange and those from detachment are colored purple. The periodic vortex shedding can be seen in the repeated development and movement downstream of saddle-center pairs.

$3D$ Separated Flows

3-D separated flows play a significant role in aerodynamics because separation surfaces are often associated with vortices and recirculation zones, which are important structures of the flow far from the body [53].

As in the 2-D example of Figures 2.8 and 2.9, the fluid in 3-D separated flows moves parallel to the body and then suddenly detaches from the surface, creating vortices in the wake. The fluid can also reattach, causing recirculation regions similar to the bubble in Figure 2.8. However, detachment and attachment in 3-D flows do not arise at isolated points on the surface of the body, but are distributed along entire

Multiple Pages Missing from Available Version

$2D$ tensor field topology and covered some work in $3D$ space. We outline some of his findings in this section.

A second-order tensor field $\mathbf{T}(\vec{x})$ defined on an open subset E of $\mathbf{R}^n$ is a mapping $\mathbf{T}$ that associates with each vector $\vec{x}$ of E a linear transformation onto itself. In Cartesian coordinates, $\mathbf{T}$ can be represented by n^2 components $T_{ij}, i, j = 1, ..., n$. And the transformation of its components follow the law:

$$T'_{ij} = \sum_{p,q=1} n\beta_{ip}\beta_{jq}T_{pq} \tag{2.5}$$

under an orthonormal transformation $\beta = \beta_{ij}$ of the coordinate systems (see Chapter 1, section 1.2.1).

The equivalent representation of a tensor field $\mathbf{T}$ is using its eigenvectors and eigenvalues:

$$\vec{v}_i = \lambda_i \vec{e}_i \tag{2.6}$$

where $i = 1, 2, ..., n$ (n is the dimension) and the eigenvalues are numbered as $\lambda_1 \geq \lambda_2 .... \geq \lambda_n$. Eigenvectors are bi-directional due to their sign indeterminacy (Figure 1.10). When the tensor components of $\mathbf{T}$ are continuous and differentiable functions of E, its eigenvectors and eigenvalues are continuous and smooth at most points (except at degenerate points).

Any tensor $\mathbf{T}$ can be decomposed into the sum of a symmetric tensor $\mathbf{S}$ and an antisymmetric tensor $\mathbf{A}$:

$$\begin{aligned} \mathbf{S} &= \frac{1}{2}(\mathbf{T} + \mathbf{T'}) \\ \mathbf{A} &= \frac{1}{2}(\mathbf{T} - \mathbf{T'}) \end{aligned}$$

where $\mathbf{T'}$ is the transpose of $\mathbf{T}$. The eigenvalues and eigenvectors of $\mathbf{S}$ are real and orthogonal. The antisymmetric tensor has only three independent components that

form a tensor known as "axial vector".[3] The focus in the following discussion will be on $2D$ symmetric tensor fields with some partial extension in $3D$ space.

2.2.1 Degenerate Point Theory

Second-order tensor fields have many properties analogous to vector fields. Degenerate points are points where at least two eigenvalues are equal to each other[4]. They are basic singularities underlying the topology of tensor fields and play a similar role as critical points in vector fields. Away from degenerate points, the hyperstreamlines[5] are smooth, continuous and non-intersecting trajectories. The tensor data in this region are a diffeomorphism to a constant field (Figure 2.12). Therefore, degenerate points and their vicinities are the only places that are topologically interesting and they determine the behavior of the whole field.

Tensor Index

Similar to vector fields, the local behavior about degenerate points can be characterized by a **tensor index**. The notion of a "tensor index of a curve" is first introduced. And the concept of "tensor index at a degenerate point" is derived from this notion which is an extension from vector field topology.

The "curve" under consideration is a Jordan curve:

[3]The antisymmetric tensor field

$$\mathbf{T} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

is equivalent to the vector field

$$\vec{v} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

in the sense that ω_1, ω_2, and ω_3 transform as the components of a vector field.

[4]see Chapter 1, section 1.2.1

[5]definition see Chapter 1, section 1.4.2

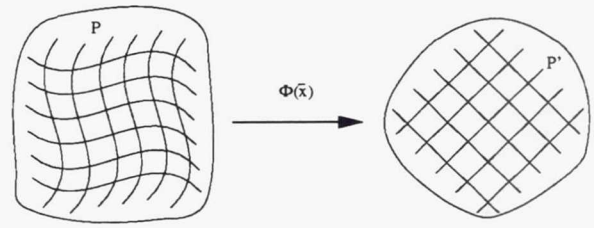


Figure 2.12: Away from degenerate points, continuous and symmetric tensor fields are diffeomorphic to constant data.

Definition 5 (Jordan Curve) *A Jordan curve is a curve which is homeomorphic to a circle—i.e., a piecewise-smooth, simple, closed curve.*

And the "tensor index of a curve" is defined as below:

Definition 6 (Tensor Index of a Jordan Curve) *Let E be an open set of $\boldsymbol{R}^2$ and let $L \subset E$ be a Jordan curve. The index $I_{\boldsymbol{T}}(L)$ of L relative to a tensor field $\boldsymbol{T} \in C^1(E)$, where $\boldsymbol{T}$ has no degenerate points on L, is the number of counterclockwise revolutions made by the eigenvectors of $\boldsymbol{T}$ when traveling once in a counterclockwise direction along L.*

The index $I_{\mathbf{T}}(L)$ can be computed as follows:

The angle α between the eigenvectors of $\mathbf{T}$ and the x-axis (Figure 2.13) is given by

$$\tan 2\alpha = \frac{2T_{12}}{T_{11} - T_{22}}$$

which implies

$$\alpha = \frac{1}{2} \arctan \frac{2T_{12}}{T_{11} - T_{22}} \tag{2.7}$$

Equation 2.7 defines α only locally and within an integral multiple of $\frac{\pi}{2}$. The differential of α, however, is a well-defined, smooth, differential form in the whole domain E:

$$d\alpha = \frac{(T_{11} - T_{22})dT_{12} - T_{12}d(T_{11} - T_{22})}{(T_{11} - T_{22})^2 + 4T_{12}^2} \tag{2.8}$$

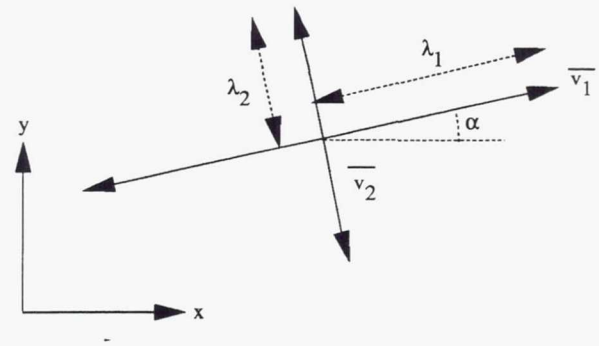


Figure 2.13: The two orthogonal eigenvector fields $\vec{v}_i$, represented as bidirectional arrows.

From Definition 6, the index $I_{\mathbf{T}}(L)$ of a Jordan curve L is equal to the line integral

$$I_{\mathbf{T}}(L) = \frac{1}{2\pi} \oint_{L^+} d\alpha$$

where L^+ means that the curve L is traversed in a counterclockwise direction. Thus,

$$I_{\mathbf{T}}(L) = \frac{1}{2\pi} \oint_{L^+} \frac{(T_{11} - T_{22})dT_{12} - T_{12}d(T_{11} - T_{22})}{(T_{11} - T_{22})^2 + 4T_{12}^2} \tag{2.9}$$

Equation 2.9 will prove extremely useful in analyzing degenerate points (Section 2.2.1).

Now a tensor index can be associated with degenerate points by enclosing them with Jordan curves.

Definition 7 (Tensor Index at a Degenerate Point) *Suppose that $\boldsymbol{T} \in C^1(E)$ where E is an open subset of $\boldsymbol{R}^2$. Let $\vec{x}_0 \in E$ be an isolated degenerate point of $\boldsymbol{T}$ and let $L \subset E$ be a Jordan curve encompassing $\vec{x}_0$ in its interior and no other degenerate points of $\boldsymbol{T}$. Then the index of $\boldsymbol{T}$ at the degenerate point $\vec{x}_0$ is defined as*

$$I_{\boldsymbol{T}}(\vec{x}_0) = I_{\boldsymbol{T}}(L)$$

The following corollary results from Definition 7:

Corollary 1 *The tensor index at a degenerate point of a tensor field $\boldsymbol{T} \in C^1(E)$, where E is an open subset of $\boldsymbol{R}^2$, is an integer or half-integer quantity*[6].

And there also exists a theorem similar to Poincaré index theorem in vector fields that deals with a Jordan curve encompassing several isolated degenerate points.

Theorem 2 *Suppose that $\boldsymbol{T} \in C^1(E)$, where E is an open subset of $\boldsymbol{R}^2$ containing a Jordan curve L. Assume that there are no degenerate points on L but that there are a finite number of degenerate points, $\vec{x}_1, \ldots, \vec{x}_n$, in the interior of L. It then follows that*

$$I_{\boldsymbol{T}}(L) = \sum_{j=1}^{n} I_{\boldsymbol{T}}(\vec{x}_j) \tag{2.10}$$

Basic Building Blocks

Hyperstreamlines in the vicinity of degenerate points also form several basic building blocks as streamlines around critical points in vector field.

Consider an isolated degenerate point $\vec{x}_0$ of a tensor field $\mathbf{T}(\vec{x})$. In most cases, the eigenvector fields in the vicinity of $\vec{x}_0$ can be described in terms of three types of angular sectors (Figure 2.14):

- hyperbolic sectors α_i, where trajectories sweep past the degenerate point;
- parabolic sectors β_j, where trajectories lead away or towards the degenerate point; and
- elliptic sectors γ_k, where trajectories begin and end at the degenerate point.

The number of hyperbolic, parabolic, and elliptic sectors are denoted as n_h, n_p, and n_e, respectively. Eigenvectors rotate in different directions, when passing through sectors of different types along a closed curve that encompasses a degenerate point.

[6] *Proof see Reference [38]*

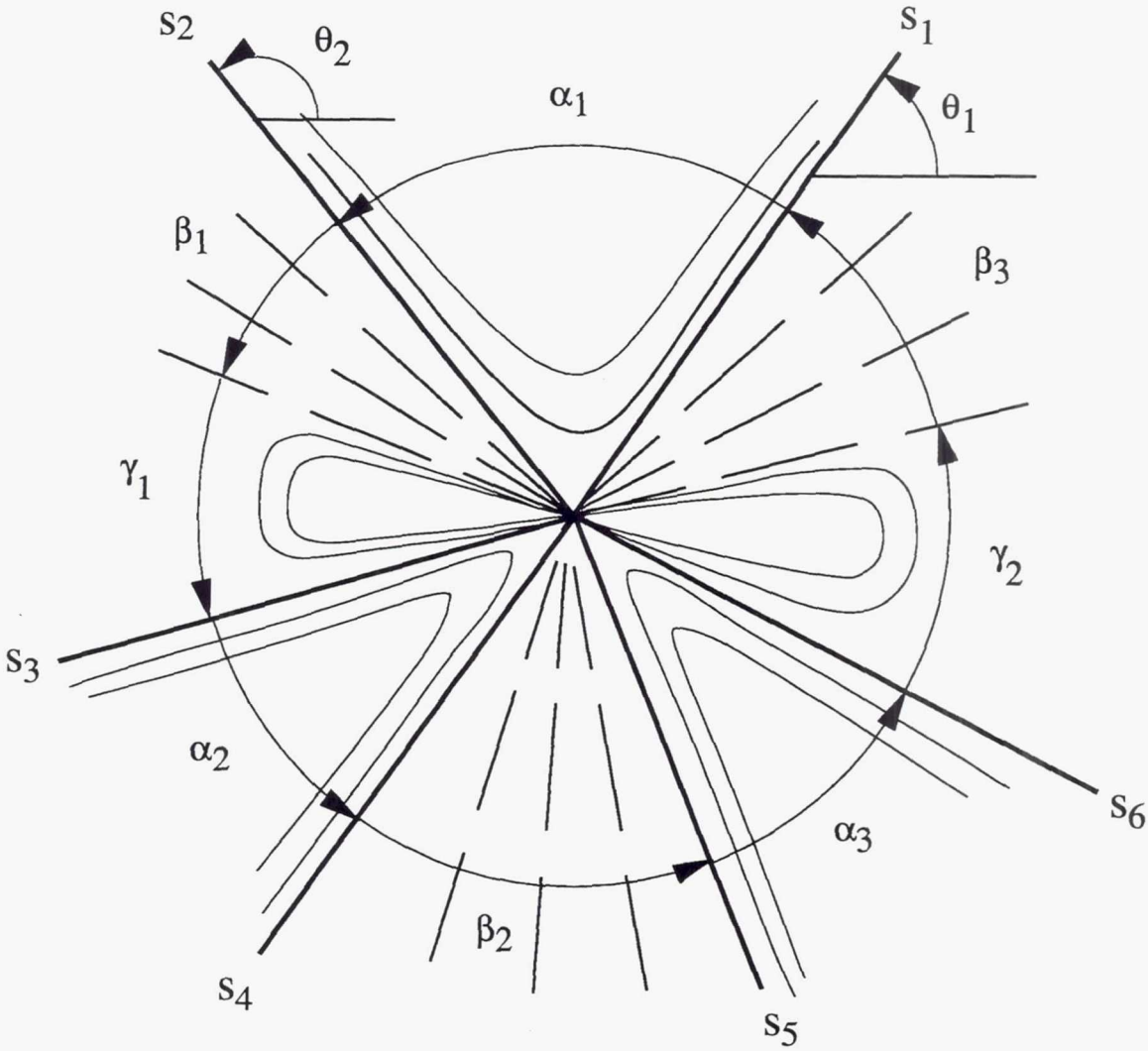


Figure 2.14: Hyperbolic (α_i), parabolic (β_j), and elliptic (γ_k) sectors at a degenerate point.

Sector	Eigenvector Rotation
parabolic	counterclockwise
elliptic	counterclockwise
hyperbolic:	
$\alpha_i > 180°$	counterclockwise
$\alpha_i < 180°$	clockwise

Table 2.1: Rotation of the eigenvectors during a counterclockwise motion around a degenerate point.

Table 2.1 shows the direction in which eigenvectors rotate during a counterclockwise motion around a degenerate point.

Most degenerate points are built from a series of adjacent hyperbolic, parabolic, and elliptic sectors. However, some degenerate points only have one type of sector. A "trisector point" (Section 3.3.3) has only hyperbolic sector and a "star-node", has only one parabolic sector. Furthermore, some specific points, namely centers and foci, have neither types of sectors. In other words, they are made up of $n_h = 0$ hyperbolic, $n_p = 0$ parabolic, and $n_e = 0$ elliptic sectors.

Hyperstreamline trajectories that separate adjacent sectors at a degenerate point play an important topological role. Then "Separatrices" of a tensor field can be defined by analogy with a similar concept from vector field topology (References [62, 63]).

Definition 8 (Separatrices) *A separatrix of a tensor field* $\boldsymbol{T}$ *is a hyperstreamline trajectory which lies on the boundary of a hyperbolic sector at a degenerate point of* $\boldsymbol{T}$.

Separatrices lie on hyperbolic/hyperbolic, hyperbolic/parabolic, and hyperbolic/elliptic boundaries, but trajectories along parabolic/parabolic, parabolic/elliptic, and elliptic/elliptic boundaries are not to be considered as separatrices. Also, adjacent parabolic sectors are always merged into a unique parabolic sector.

$2D$ Degenerate Points

For a $2D$ symmetric tensor field,

$$\begin{pmatrix} T_{11}(\vec{x}) & T_{12}(\vec{x}) \\ T_{12}(\vec{x}) & T_{22}(\vec{x}) \end{pmatrix}$$

degenerate points satisfy the conditions:

$$\begin{cases} T_{11}(\vec{x_0}) - T_{22}(\vec{x_0}) & = & 0 \\ T_{12}(\vec{x_0}) & = & 0 \end{cases}$$

The vicinity of a degenerate point can be expressed as:

$$\begin{aligned} \tfrac{T_{11}-T_{22}}{2} &\approx a(x-x_0) + b(y-y_0) \\ T_{12} &\approx c(x-x_0) + d(y-y_0) \end{aligned} \tag{2.11}$$

where $\vec{x_0}$ is the location of the degenerate point and

$$\begin{aligned} a &= \tfrac{1}{2}\tfrac{\partial(T_{11}-T_{22})}{\partial x}\Big|_{\vec{x}_0} & b &= \tfrac{1}{2}\tfrac{\partial(T_{11}-T_{22})}{\partial y}\Big|_{\vec{x}_0} \\ c &= \tfrac{\partial T_{12}}{\partial x}\Big|_{\vec{x}_0} & d &= \tfrac{\partial T_{12}}{\partial y}\Big|_{\vec{x}_0} \end{aligned} \tag{2.12}$$

An important quantity for characterizing degenerate points is

$$\delta = ad - bc \tag{2.13}$$

where δ is invariant under rotation.

For simple degenerate points ($\delta \neq 0$) at $\vec{x_0}$, the tensor index is defined as:

$$I_{\mathbf{T}}(\vec{x}_0) = \frac{1}{2} sign(\delta) = \pm\frac{1}{2} \tag{2.14}$$

It follows that there are only two patterns possible for simple degenerate points (Figure 2.15):

- Trisector point: $\delta < 0$, $I_{\mathbf{T}}(\vec{x}_0) = -\frac{1}{2}$ and $n_h = 3$, $n_e = 0$ and $n_p = 0$,
- Wedge point: $\delta > 0$, $I_{\mathbf{T}}(\vec{x}_0) = \frac{1}{2}$ and $n_h = 1$, $n_e = 0$ and $n_p = 1$ (special case when two separatrices are equal, then $n_p = 0$)

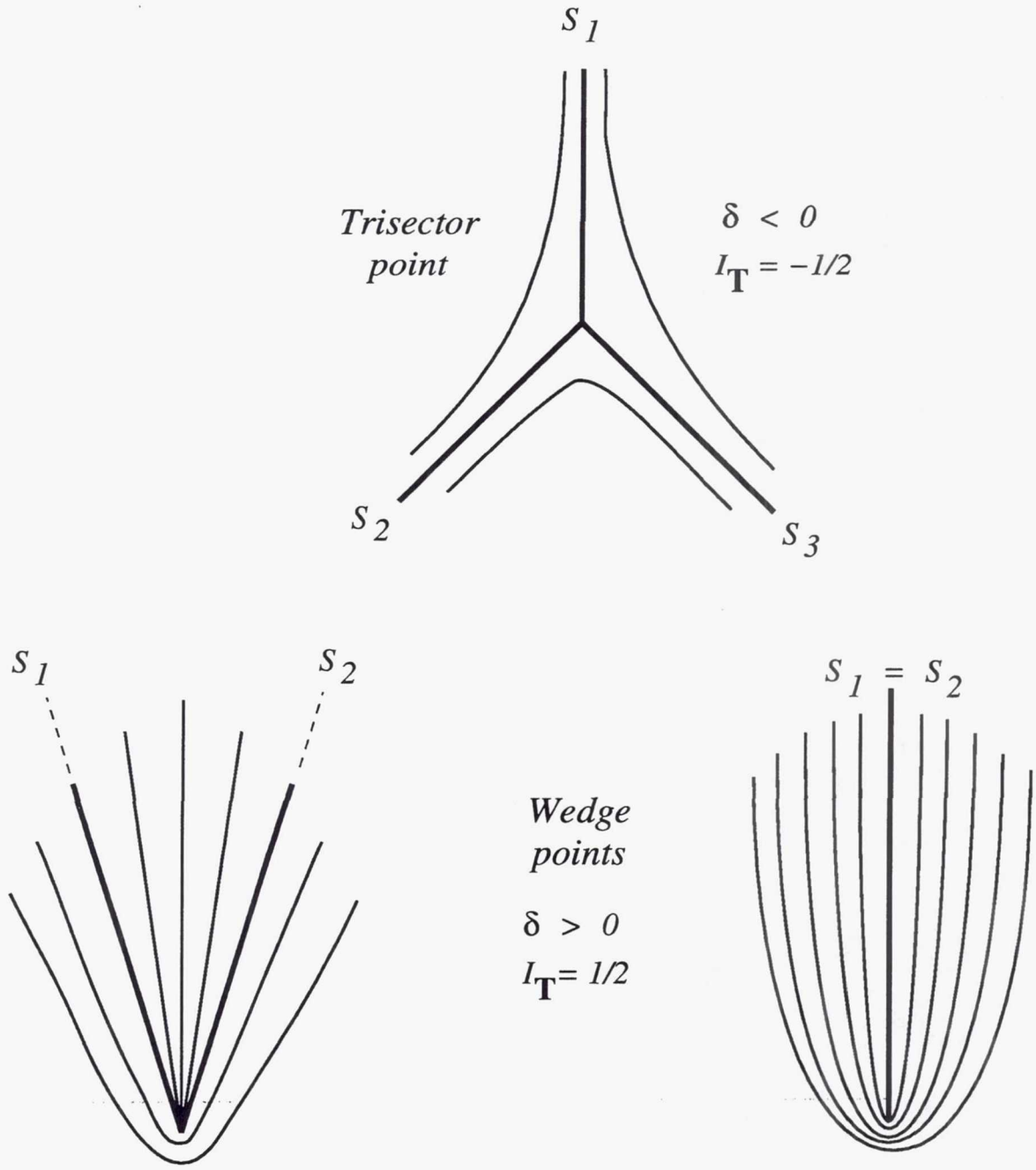


Figure 2.15: Simple degenerate points. $\delta = ad - bc \neq 0$ and $I_{\mathbf{T}}$ = tensor index. Trisector ($\delta < 0$) and wedge ($\delta > 0$) points. Trajectories s_k are separatrices.

Extensions to $3D$ Degenerate Points

In this section, some extensions of the theory of degenerate points to $3D$ space will be discussed. In this case, there are three real eigenvalues[7], λ_i, and three orthonormal eigenvectors, $\vec{e}_i$, $i = 1, 2, 3$, at each point of space.

$$\vec{v}_i(\vec{x}) = \lambda_i(\vec{x})\vec{e}_i(\vec{x}) \qquad \text{i} = 1,\ 2,\ 3.$$

Various types of degenerate points $\vec{x}_0$ exist:
A double degenerate point:

$$\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0), \tag{2.15}$$

$$\lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0), \tag{2.16}$$

and a triple degenerate point:

$$\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0), \tag{2.17}$$

respectively.[8]

Consider double degenerate case where two eigenvalues are identical —for example, $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) > \lambda_3(\vec{x}_0)$. The tensor field is degenerate in the plane orthogonal to $\vec{v}_3(\vec{x}_0)$ within which locally two-dimensional patterns such as wedge points, trisectors, and multiple degenerate points can occur. Figure 2.16 represents the tensor field around a wedge point and a trisector in the plane orthogonal to $\vec{v}_3(\vec{x}_0)$.

In fact, the patterns in Figure 2.16 should be drawn on a surface normal to the non-degenerate eigenvector field, $\vec{v}_3(\vec{x})$. However, when close enough to $\vec{x}_0$, this surface can be approximated with arbitrary precision by its tangent plane $\mathbf{P}$ at $\vec{x}_0$. (Away from $\vec{x}_0$, trajectories depart from $\mathbf{P}$.)

Let $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda$, $\mathbf{B}^0 = \{\beta^0_{ij}\}$, $i, j = 1, 2, 3$, be the rotation matrix that brings the third axis of the coordinate system along $\vec{e}_3(\vec{x}_0)$ —the unit eigenvector

[7] Here, we only study symmetric tensors.
[8] Recall that $\lambda_1(\vec{x}) \geq \lambda_2(\vec{x}) \geq \lambda_3(\vec{x})$ at every point $\vec{x}$.

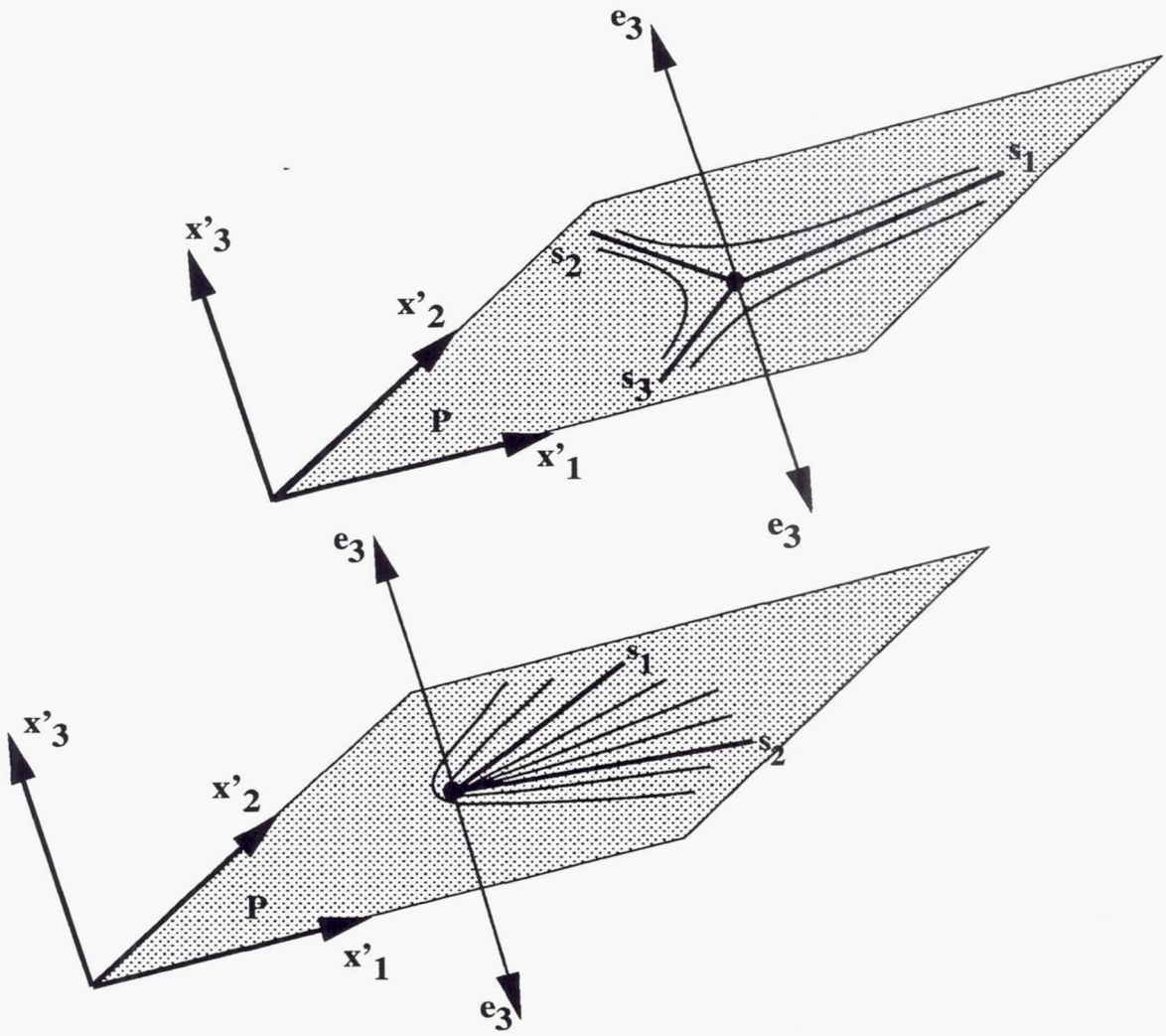


Figure 2.16: Three-dimensional wedge point and trisector around a degenerate point where $\lambda_1 = \lambda_2 > \lambda_3$.

associated to $\lambda_3(\vec{x}_0)$ (Figure 2.16). Transforming $\mathbf{T}$ by the rotation $\mathbf{B}^0 = \{\beta^0_{ij}\}$ leads to new components

$$T'_{ij} = \sum_{p,q=1}^{3} \beta^0_{ip}\beta^0_{jq}T_{pq}$$

At $\vec{x}_0$, $\mathbf{T}$' is diagonalized—i.e.,

$$\mathbf{T}'(\vec{x}_0) = \{T'_{ij}(\vec{x}_0)\} = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

Let $x'_i = \sum_{p=1}^{3} \beta^0_{ip}(x_p - x^0_i)$, $i = 1, 2, 3$, be the new coordinates. $x'_i = 0$ at the location of the degenerate point $\vec{x}_0$. x'_1 and x'_2 run along the plane $\mathbf{P}$ while x'_3 goes in the direction normal to $\mathbf{P}$. This new frame of reference is shown in Figure 2.16, where we translated the origin away from the degenerate point for clarity. The components of $\mathbf{T}$' in the vicinity of $\vec{x}_0$ can be expanded to first-order terms as

$$T'_{ij}(\vec{x}') \approx T'_{ij}(\vec{x}_0) + \sum_{k=1}^{3} \alpha'_{ijk}x'_k$$

with $i, j, k = 1, 2, 3$ and $\alpha'_{ijk} = \frac{\partial T'_{ij}}{\partial x'_k}$. Tensor patterns can be analyzed in the plane $\mathbf{P}$ (Figure 2.16) by letting $x'_3 \equiv 0$ and considering the 2×2 block

$$\begin{pmatrix} T'_{11} & T'_{12} \\ T'_{12} & T'_{22} \end{pmatrix} = \begin{pmatrix} \lambda + \alpha'_{111}x'_1 + \alpha'_{112}x'_2 & \alpha'_{121}x'_1 + \alpha'_{122}x'_2 \\ \alpha'_{121}x'_1 + \alpha'_{122}x'_2 & \lambda + \alpha'_{221}x'_1 + \alpha'_{222}x'_2 \end{pmatrix}$$

The important parameter is $\delta = a'd' - b'c'$, where $a' = \frac{1}{2}(\alpha'_{111} - \alpha'_{221})$, $b' = \frac{1}{2}(\alpha'_{112} - \alpha'_{222})$, $c' = \alpha'_{121}$, and $d' = \alpha'_{122}$. Trisectors, wedge points, and multiple degenerate points correspond to $\delta < 0$, $\delta > 0$, and $\delta = 0$, respectively.

The discussion above concerns degenerate points where $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) > \lambda_3(\vec{x}_0)$. The case of degenerate points characterized by $\lambda_1(\vec{x}_0) > \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0)$ is similar. However, fully three-dimensional patterns that exist in the vicinity of degenerate points where $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0) \equiv \lambda$ will be analyzed in Chapter 4.

2.2.2 Global Topology

Tensor field topology describes the behavior of a collection of an infinite number of hyperstreamlines in the field. The basic constituents of second-order tensor field topology are

- the degenerate points (Section 2.2.1) and
- the set of their connecting hyperstreamlines (separatrices).

The technique to extract the topology of tensor fields and to study topological transitions is similar to that of vector fields. Tensor fields are visualized by presenting their eigenvector fields individually. Each eigenvector field is represented by a topological skeleton obtained by locating degenerate points and integrating the set of their connecting separatrices.

$2D$ Time-Dependent Fields

In this section, an example of a stress tensor field in a $2D$ flow past a cylinder is illustrated. The purpose of visualizing this field is to study its topological transitions. Fluid elements undergo compressive stresses while moving with the flow. Stresses are described mathematically by the stress tensor, which combines isotropic pressure and anisotropic viscous stresses.

$$\sigma = -p\delta_{ij} + \nu\epsilon_{ij} + \lambda\delta_{ij}\frac{\partial v_k}{\partial x_k}$$

where $\epsilon_{ij} = \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}$, $i, j = 1, 2, 3$, $p =$ pressure, v_i and $v_j =$ velocity, and ν and $\lambda =$ viscosity coefficients.

Both eigenvalues of the stress tensor are negative and the two orthogonal eigenvectors, $\vec{v}_1$ and $\vec{v}_2$ (Equation 2.6), are along the least and the most compressive directions, respectively. At a degenerate point, the viscous stresses vanish and both eigenvalues are equal to the pressure; degenerate points are points of pure pressure.

The first step for understanding the structure of the stress-tensor field consists in tracking the motion of degenerate points over time. Degenerate points are located at each time step.

Figure 2.17 shows results at a specific instant in time. The dots mark the location of simple degenerate points. The top view shows the overall data while the bottom view focuses on the region close to the body. The colored background encodes the magnitude λ_2 of the most compressive force, from very compressive (red), to mildly compressive (orange, yellow, green), to little compressive (blue). Wedge points and trisectors are represented as black and white dots, respectively. It is shown clearly that wedge points are situated in the wake about the cylinder axis while trisector points are located off-axis. The images in Figure 2.17 belong to two video clips that visualize the motion of degenerate points over time.

Topological Skeletons

Topological skeletons are obtained by detecting degenerate points and integrating the set of their connecting separatrices. Trisector points in tensor fields play the topological role of saddle points in vector fields. As shown in Figure 2.18, they deflect adjacent trajectories in any one of their three hyperbolic sectors toward topologically distinct regions of the domain. Wedge points possess both a hyperbolic and a parabolic sector. They deflect trajectories adjacent in their hyperbolic sector, and terminate trajectories impinging on their parabolic sector. The texture represents the most compressive eigenvector of the stress tensor ($\vec{v}_2$). Color encodes as before the magnitude of the compressive force (λ_2), from most compressive (red) to least compressive (blue). The structure of the tensor field is illustrated by superimposing the topological skeleton of $\vec{v}_2$.

The orientation of the eigenvectors at any point in the plane from the topological skeleton can be inferred as follows: hyperstreamlines curve their trajectories so as to

Multiple Pages Missing from Available Version

follow the shape of the separatrices, bending around wedge points. In Figure 2.18 for example, wedge points are located around the cylinder's axis and trisectors quickly move off-axis. The separatrices that emanate from the trisector points delineate the global structure of the most compressive eigenvector; hyperstreamlines oscillate about the cylinder's axis, presenting a distorted, wave-like pattern with sharp turns at the wedge points. [9]

2.3 Chapter Summary

This chapter is an overview of vector and second-order tensor field topology. It presents the state of art in the theory of vector and second-order tensor field topology. Vector field topology has been studied for many years dating back to 19th century while second-order tensor field topology has not been studied until recently due to its complexity.

Vector fields are characterized by their critical points and the set of separatrices connecting these points. In the vicinity of a critical point, the local topological structure is determined by the Jocobian matrix and classified primarily by the Poincaré index. There are six basic patterns of a simple critical point in a vector field:an attracting/repelling node, an attracting/repelling focus, a center, or a saddle. By integrating the separatrices emanating from critical points, a topological skeleton can be constructed to depict the global topology of a vector field. Topology theory in $3D$ space can be studied similarly. The critical points in $3D$ can be classified in its three eigenvector planes. Applications to $2D$ time-dependent flow and $3D$ separated flows show that visualization technique such as streamlines, streamsurfaces combined with topology can capture the key properties such as vortices as well as provide a very insightful picture of a vector field. Yet, an important subject as how to quantitatively

[9] A video of this animated flow can be requested from Prof. Lambertus Hesselink

measure the closeness between different vector fields has not been addressed. This study will be covered in Chapter 3.

Second-order tensor fields have many properties analogous to vector fields. Degenerate points are basic topological elements and play a similar role as critical points in vector fields. In the vicinity of a degenerate point, the local structure is determined by its invariant δ. Based on the Poincaré index, the concept of a tensor index is developed and classifies the degenerate points. Because of the sign indeterminacy, the simple degenerate points in $2D$ only have two basic patterns: a trisector point and a wedge point. The pattern of a double degeneracy in $3D$ can be studied similarly after an appropriate rotation in space. A topological skeleton of a tensor field can be obtained by connecting degenerate points with their separatrices. It depicts the global topology structure of a tensor field. An example of a $2D$ stress field in a time-varying flow is given; the global structure and its transition are shown. Topics concerning topology in $3D$ space, like the structure of a triple degenerate point, the importance of a deviator and $3D$ representation of non-isolated degenerate points, will be discussed in Chapter 4.

Chapter 3

Feature comparisons of vector fields using Earth Mover's Distance

Vector fields have numerous applications in physical sciences. The concept of critical points in vector fields has been well studied and has proven to play a key role in vector field topology. However, a formal study on the quantitative measurement of similarities and differences between vector fields using topology has not been done. In fluid dynamics, simplifications are necessary in the numerical simulation process in order to compute analytical solutions of flows over complex bodies in a reasonable amount of time. Researchers formulate various models with the hope of capturing the essential features of real flows. For example, Reynolds stress and velocity fields in turbulent flows are almost impossible to solve analytically. Many turbulence models, both linear and nonlinear, have been suggested to better capture real flows (Chapter 1, section 1.3.3). However, the models for a large part have been quite arbitrary. In order to test the validity of a certain model, a comparison between the computational and experimental results are necessary. However, a metric for the closeness between the computational and experimental results remains unsolved. Traditionally, a series of charts for each component of the Reynolds stress and velocity vector has

been drawn. It is perceptually nonintuitive as well as difficult to show the intrinsic structural differences between the two fields. Past study in our group has focused on the geometric structure of vector and tensor fields [9] [38] [15]. However, a topological property of a vector field is a property possessed by this vector field and all its topological equivalents. The dissimilarity of their appearances sometimes fools the human visual system. For example, vector fields 15 and 16 in Figure 3.10, are two swirling flows that have the same topological structure yet appear quite different to a human observer. Therefore, a quantitative measurement for comparison of vector fields is essential.

Earth Mover's Distance (EMD), first introduced by Yossi Rubner et al. for fast retrieval of similar images in a large data base [64] [65], computes the minimal amount of work that must be performed to transform one distribution into the other by moving "distribution mass" around. In our vector field analysis, the feature distribution is defined as the set of critical points and their corresponding α, β parameters (to be defined later) of a vector field. After extracting the feature distribution, a set of EMDs are computed. Multidimensional Scaling (MDS) was originally used for color perception and spatial frequency discrimination [66]. Here, with the help MDS, we can display the similarities and differences between selected vector fields.

3.1 Clifford Algebra

In 1876, Clifford introduced "geometric algebras" [67] which is now known as "Clifford algebras". It is a generalization of Grassman's exterior algebra and Hamilton's quaternions, both of which sought to capture the geometric and algebraic properties of Euclidean space [68].

Roughly speaking, a Clifford algebra is an associative algebra with unit element 1 into which a given Euclidean or Minkowski space may be embedded, in which the

corresponding quadratic form may be expressed as the negative of a square. By using Clifford algebra, the geometry of a complex field can be described as a multiplication of simple vector fields.

Definitions

Definition 9 (Quadratic Form) *Let V be a finite-dimensional vector space over the scalar field F, where $F = R$ or C. A quadratic form on V is a mapping $Q : V \to F$ such that*

$$Q(\lambda v) = \lambda^2 Q(v), \quad \lambda \in F, \quad v \in V,$$

and the associated form

$$B(v, w) = \tfrac{1}{2}\{Q(v) + Q(w) - Q(v - w)\}, \qquad v, w \in V,$$

is bilinear.

Definition 10 (Quadratic Space) *When a quadratic form Q on V exists, the pair (V, Q) is said to be a quadratic space.*

Definition 11 (Clifford Algebra) *Let (V, Q) be an arbitrary quadratic space, $\boldsymbol{A}$ be an associate algebra over F with identity 1 and ν: $V \to \boldsymbol{A}$ an F-linear embedding of V into $\boldsymbol{A}$. The pair $(\boldsymbol{A}, \nu)$ is said to be a Clifford algebra for (V, Q) when*
(i) $\boldsymbol{A}$ is generated as an algebra by $\nu(v)$: $v \in V$ and λ_1: $\lambda \in F$,
(ii) $(\nu(v))^2 = -Q(v)1$, all $v \in V$.

Remark: condition (i) is a minimality restriction on the 'size' of **A** and condition (ii) ensures that **A** is an algebra in which there exists a 'square root' of the quadratic form -Q [68].

These definitions are basic to construct a "geometric algebra" for vector field analysis. For a usual $2D$ vector space V, a four-dimensional algebra G_2 can be

constructed with the basis as:

$$1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad i = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad e_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The rules of multiplication are:

$$1e_j = e_j 1 = e_j \qquad j = 1, 2 \tag{3.1}$$

$$1^2 = e_j^2 = 1 \qquad j = 1, 2 \tag{3.2}$$

$$i^2 = -1 \tag{3.3}$$

$$e_1 e_2 = e_2 e_1 = i \tag{3.4}$$

This kind of construction can be extended to any dimension n, and details can be seen in References [68, 69].

3.2 Vector field representations using Clifford Algebra

In [70, 71], Sheuermann et al. introduced Clifford vector fields which represent vector fields in the four-dimensional algebra G_2 introduced in Section 3.1. The usual vectors $(x, y) \in R^2$ are identified with

$$xe_1 + ye_2 \in E^2 \subset G_2$$

and the complex numbers $a + bi \in C$ with

$$a1 + bi \in G_2$$

For a real vector field

$$v : R^2 \to R^2$$

$$(x, y) \mapsto (v_1, v_2)$$

the clifford vector field is set as

$$\vec{E} : E^2 \to E^2$$

$$\vec{r} = xe_1 + ye_2 \mapsto v_1 e_1 + v_2 e_2$$

A Clifford vector field is just a multivector field with values in $R^2 \subset G_2$

$$v : R^2 \to R^2 \subset G_2 \tag{3.5}$$

Let $z = x + iy$, $\bar{z} = x - iy$ be complex numbers in the Clifford algebra. This means

$$x = \frac{1}{2}(z + \bar{z}) \tag{3.6}$$

$$y = \frac{1}{2i}(z - \bar{z}) \tag{3.7}$$

We get

$$\begin{aligned} \vec{v}(r) &= v_1(x, y)e_1 + v_2(x, y)e_2 \\ &= v_1\left(\frac{1}{2}(z + \bar{z}), \frac{1}{2i}(z - \bar{z})\right)e_1 \\ &\quad -iv_2\left(\frac{1}{2}(z + \bar{z}), \frac{1}{2i}(z - \bar{z})\right)e_1 \\ &= E(z, \bar{z})\, e_1 \end{aligned} \tag{3.8}$$

Generally, a linear vector field can easily be shown to be:

$$\begin{aligned} \vec{v}(r) &= E(z, \bar{z})e_1 \\ &= (az + b\bar{z} + c)e_1 \end{aligned} \tag{3.9}$$

where $a, b, c \subset C$.

Let $E : C^2 \to C$ be a complex polynomial so that $\vec{v} = E(z, \bar{z})e_1$. Let $F_k : C^2 \to C, k = 1, \ldots, n$ be the irreducible components of E, so that $E(z, \bar{z}) = \prod_{k=1}^{n} F_k$, then an arbitrary polynomial vector field with isolated critical points can be expressed as:

$$\begin{aligned} \vec{v}(r) &= E(z, \bar{z})e_1 \\ &= \prod_{k=1}^{n} (a_k z + b_k \bar{z} + c_k)e_1 \end{aligned} \tag{3.10}$$

where z_k is the unique zero of $a_k z + b_k \bar{z} + c_k$.

3.3 α-β space analysis

In Chapter 2, six basic patterns of simple critical points are discussed. They are determined by the eigenvalues of their Jacobian matrix. More convenient parameters based on eigenvalues will be introduced in the following section and they actually lead to the discovery of two more basic patterns.

3.3.1 Introduction to α-β space

For a linear vector field $\vec{v} = (az + b\bar{z} + c)e_1$, let $a = a_1 + a_2 i$ and $b = b_1 + b_2 i$. Eigenvalues of the Jacobian around its critical point z_0 are

$$\lambda_1 = b_1 + \sqrt{|a|^2 - b_2^2}$$

and

$$\lambda_2 = b_1 - \sqrt{|a|^2 - b_2^2}$$

Define two new parameters:

$$\alpha = b_1 \tag{3.11}$$

$$\beta = sign(|a|^2 - b_2^2)\sqrt{||a|^2 - b_2^2|} \tag{3.12}$$

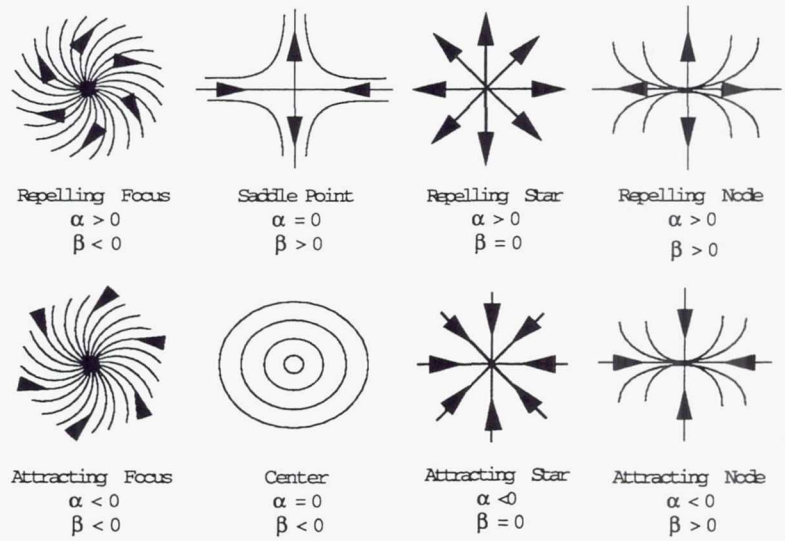


Figure 3.1: 8 basic patterns for simple critical points

Criteria for basic patterns of simple critical points are:

$$
\begin{aligned}
\beta < 0, \alpha = 0 \quad &\Longleftrightarrow \quad C \\
\beta < 0, \alpha > 0 \quad &\Longleftrightarrow \quad RF \\
\beta < 0, \alpha < 0 \quad &\Longleftrightarrow \quad AF \\
\beta = 0, \alpha < 0 \quad &\Longleftrightarrow \quad AS \\
\beta = 0, \alpha > 0 \quad &\Longleftrightarrow \quad RS \\
\beta > 0, |\beta| > |\alpha| \quad &\Longleftrightarrow \quad Sa \\
|\beta| < |\alpha| : & \\
\beta > 0, \alpha > 0 \quad &\Longleftrightarrow \quad RN \\
\beta > 0, \alpha < 0 \quad &\Longleftrightarrow \quad AN
\end{aligned}
$$

where C stands for center, RN for node, AN for attracting node, RF for repelling focus, AF for attracting focus, St for star and Sa for Saddle, see Figure 3.1. Compared to Figure 2.6, two more patterns: attracting and repelling stars are added. After careful study, it is found that the original six patterns only cover cases for which

$\beta < 0$ and $\beta > 0$ with $\beta = 0$ left out. The $\beta = 0$ case is proved to be important in order to get a smooth transition between different vector fields (see section 3.3.2)[1].

The usefulness of α and β parameters is not limited to the classification of critical points. They are essential in vector field comparisons (Section 3.4). Therefore, a new space $\alpha - \beta$ is introduced with α and β being abscissa and ordinate, respectively. It is defined as a positive-definite quadratic space with norm

$$|a| = \sqrt{a \cdot a}$$

for all $a \in \alpha - \beta$. And the distance of a from b (a and b are two points in $\alpha - \beta$.) is

$$|a - b| = \sqrt{(a - b) \cdot (a - b)}$$

The basic patterns are now points located at (α, β) in this new space. This is a true metric with normal Euclidean distance and vectors in this space obey all the rules defined for a regular $2D$ Euclidean space.

3.3.2 Circular relationship between eight basic patterns for simple critical points

Lemma 1 *Given a dynamic system governed by a vector field* $(F(x, y), G(x, y))$,

$$\begin{aligned} \frac{dx}{dt} &= F(x, y) \\ \frac{dy}{dt} &= G(x, y) \end{aligned} \tag{3.13}$$

Let $\vec{x}_0$ *be a simple critical point in the field, and* λ_1 *and* λ_2 *be the eigenvalues of the Jacobian matrix of* $\vec{x}_0$.

$$\boldsymbol{J} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

[1]There are actually a family of logarithmic node corresponding to $\beta = 0$ case, but the choice of μ (see section 3.3.2) is rather random, so we just use $\mu = 0$ as a representative here

where

$$a = \frac{\partial F(x,y)}{\partial x}\Big|_{\vec{x}_0}, \quad b = \frac{\partial F(x,y)}{\partial y}\Big|_{\vec{x}_0}$$
$$c = \frac{\partial G(x,y)}{\partial x}\Big|_{\vec{x}_0}, \quad d = \frac{\partial G(x,y)}{\partial y}\Big|_{\vec{x}_0}$$

Then: 1) if λ_1 and λ_2 are real and distinct, i.e. $\lambda_1 = \alpha + \beta$ and $\lambda_2 = \alpha - \beta$ (for definitions of α and β see section 3.3.1), there exists a nonsingular real transformation reducing system 3.13 to the canonical form

$$\begin{cases} \frac{dx}{dt} & \approx & \alpha x + \beta y \\ \frac{dy}{dt} & \approx & \beta x + \alpha y \end{cases}$$

2) if $\lambda_1 = \lambda_2$, i.e α is arbitrary and $\beta = 0$, there exists a real transformation reducing system 3.13 to the canonical form [62]

$$\begin{cases} \frac{dx}{dt} & \approx & \alpha x \\ \frac{dy}{dt} & \approx & \mu x + \alpha y \end{cases}$$

where $\mu = 0$ if $b = 0$ and $c = 0$ in system 3.13, and $\mu \neq 0$ if b and c do not vanish at the same time, in which case μ may be any number.

3) if λ_1 and λ_2 are complex, i.e. $\lambda_1 = \alpha + i\beta$ and $\lambda_2 = \alpha - i\beta$ (α arbitrary, $\beta \neq 0$), there exists a real transformation reducing system 3.13 to the canonical form [62]

$$\begin{cases} \frac{dx}{dt} & \approx & \alpha x - \beta y \\ \frac{dy}{dt} & \approx & \beta x + \alpha y \end{cases}$$

Proof: Given a dynamic system 3.13, in the neighborhood of a simple critical point $\vec{x}_0$, the vector field can be approximated as a linear field with

$$F(x, y) \approx ax + by$$

and

$$G(x, y) \approx cx + dy$$

Let the transformation matrix be:

$$S = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}$$

The new Jacobian is:

$$\mathbf{J'} = S\mathbf{J}S^{-1} = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}$$

For case 1),

$$\mathbf{J'} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Because $\mathbf{J'} = S\mathbf{J}S^{-1} \Longrightarrow \mathbf{J'}S = S\mathbf{J}$, then

$$\begin{pmatrix} \lambda_1 p_{11} & \lambda_1 p_{12} \\ \lambda_2 p_{21} & \lambda_2 p_{22} \end{pmatrix} = \begin{pmatrix} ap_{11} + cp_{12} & bp_{11} + dp_{12} \\ ap_{21} + cp_{22} & bp_{21} + dp_{22} \end{pmatrix}$$

Equate left-hand side to right-hand side,

$$p_{11}(a - \lambda_1) + p_{12}c = 0, \qquad p_{11}b + p_{12}(d - \lambda_1) = 0 \tag{3.14}$$

and

$$p_{21}(a - \lambda_2) + p_{22}c = 0, \qquad p_{21}b + p_{22}(d - \lambda_2) = 0 \tag{3.15}$$

If both b and c are zero, then the original system is already in its canonical form; otherwise, assume $c \neq 0$ and the nontrivial solution takes the form:

$$\frac{p_{12}}{p_{11}} = \frac{\lambda_1 - a}{c}, \qquad \frac{p_{22}}{p_{21}} = \frac{\lambda_2 - a}{c} \tag{3.16}$$

Clearly, the determinant of S is not zero and is a nonsingular transformation. Because $\lambda_1 = \alpha + \beta$ and $\lambda_2 = \alpha - \beta$, Equation 3.16 can be rewritten as:

$$\frac{p_{12}}{p_{11}} = \frac{\alpha + \beta - a}{c}, \qquad \frac{p_{22}}{p_{21}} = \frac{\alpha - \beta - a}{c} \tag{3.17}$$

Choose $p_{11} = p_{21} = c$, and x and y can be transformed into $X = p_{11}x + p_{12}y$ and $Y = p_{21}x + p_{22}y$. Then $\frac{dX}{dt} = \lambda_1 X$ and $\frac{dY}{dt} = \lambda_2 Y$ Now let $X = u + v$ and $Y = u - v$, then

$$\begin{aligned}
\frac{du}{dt} &= \frac{1}{2}(\frac{dX}{dt} + \frac{dY}{dt}) \\
&= \frac{1}{2}(\lambda_1 X + \lambda_2 Y) \\
&= \frac{1}{2}[(\alpha + \beta)(u + v) + (\alpha - \beta)(u - v)] \\
&= \frac{1}{2}(2\alpha u + 2\beta v) \\
&= \alpha u + \beta v \qquad (3.18)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\frac{dv}{dt} &= \frac{1}{2}(\frac{dX}{dt} - \frac{dY}{dt}) \\
&= \frac{1}{2}(\lambda_1 X - \lambda_2 Y) \\
&= \frac{1}{2}[(\alpha + \beta)(u + v) - (\alpha - \beta)(u - v)] \\
&= \frac{1}{2}(2\beta u + 2\alpha v) \\
&= \beta u + \alpha v \qquad (3.19)
\end{aligned}$$

This new system is in the canonical form and is transformed from the original system by a matrix

$$\begin{pmatrix} c & \alpha - a \\ 0 & \beta \end{pmatrix}$$

with $determinant = \beta c \neq 0$.

Case 3) can be proved similarly except $\lambda_1 = \alpha + i\beta$, $\lambda_2 = \alpha - i\beta$, and let $X = u + iv$, $Y = u - iv$.

Case 2) needs some special care. Since $\lambda_1 = \lambda_2 = \alpha$, it is in its canonical form if both b and c vanish, and $\mu = 0$. Now suppose that b and c are not both zero, say

$c \neq 0$, then from the first equation of (3.16),

$$\frac{p_{12}}{p_{11}} = \frac{\lambda_1 - a}{c}$$

and there exists no p_{ij} satisfying equation 3.14 with nonvanishing determinant. Thus choose $Y = \mu y$ with μ a random real number, then the new system is

$$\begin{aligned} \frac{dX}{dt} &= \alpha X \\ \frac{dY}{dt} &= \mu X + \alpha Y \end{aligned} \tag{3.20}$$

with a nonsingular transformation matrix

$$\begin{pmatrix} c & \alpha - a \\ 0 & \mu \end{pmatrix}$$

The trajectory here is a logarithmic node with a phase portrait satisfying $y = \frac{\mu}{\alpha} x \ln(cx)$ where c is an integration constant.

This completes the proof. $\diamond$

Lemma 1 shows that as μ goes from $-\infty$ to 0 to ∞, there are actually a family of logarithmic nodes ($\mu = 0$ is a special case and the pattern is a star) corresponding to the same set of α and β with $\beta = 0$ (Figure 3.2). If $\mu = 0$, the original system is in its canonical form and the pattern is determined to be a star. If the original system is not in its canonical form, then $\mu \neq 0$ and is arbitrary, we will choose $\mu = \alpha$ for convenience. Also, the choice of μ does not affect the values of α and β.

Theorem 3 *The ratio* $\boldsymbol{R} = \frac{\beta}{\alpha}$ *determines the local topological structure of a simple critical point.*

Proof: The proof is divided into three parts corresponding to the three cases in *Lemma* 1.

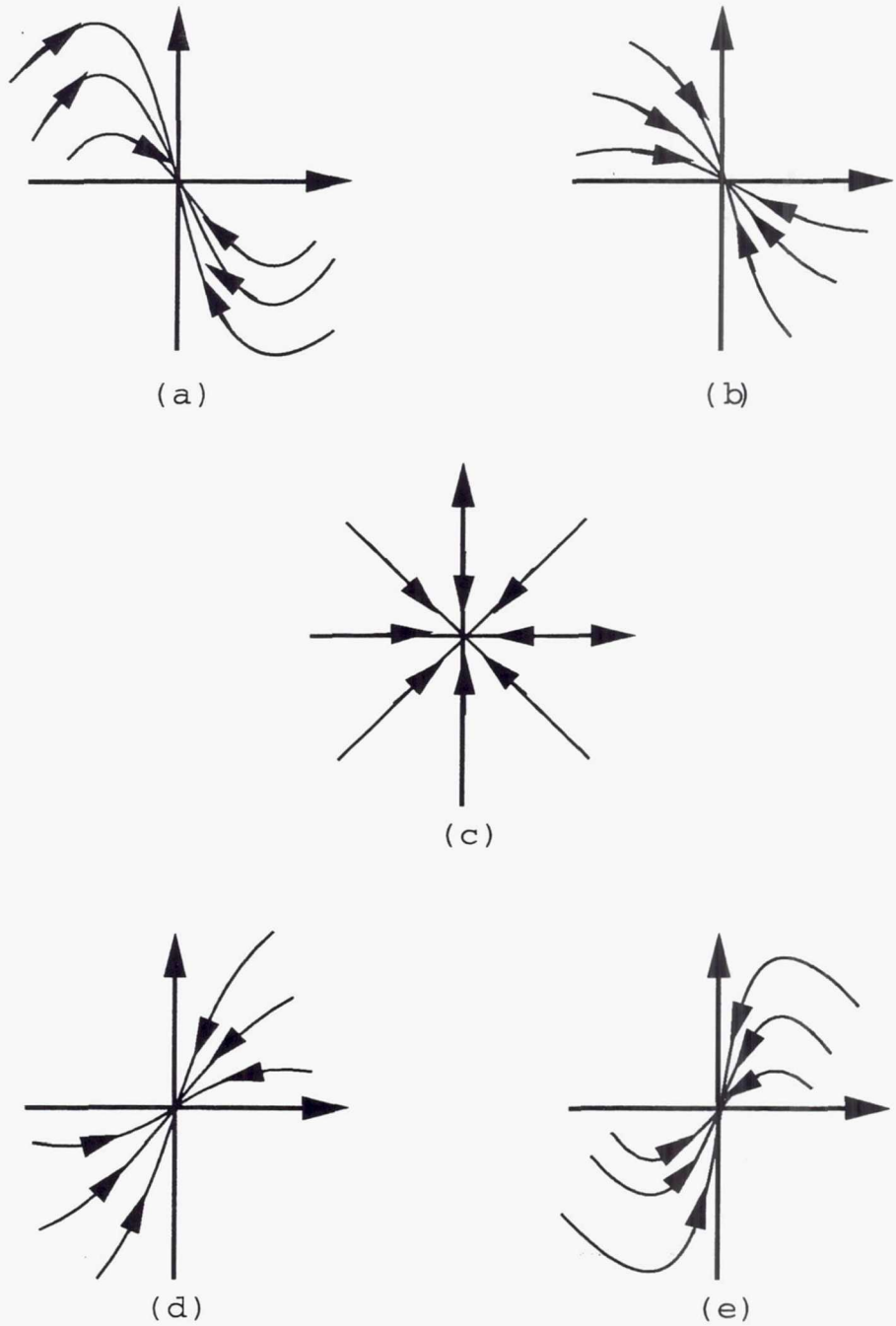


Figure 3.2: From (a)-(e), μ decreases from positive (a), (b) to zero (c) to negative (d), (e)

Case 1) λ_1 and λ_2 are real and distinct, then the system can be reduced to its canonical form:

$$\begin{cases} \frac{dX}{dt} &= \alpha X + \beta Y \\ \frac{dY}{dt} &= \beta X + \alpha Y \end{cases}$$

The trajectory can be obtained by solving function:

$$\frac{dX}{dY} = \frac{\alpha X + \beta Y}{\beta X + \alpha Y} \tag{3.21}$$

For $\alpha \neq 0$, divide both the numerator and the denominator on the right hand side in equation 3.21 by α,

$$\frac{dX}{dY} = \frac{X + \mathbf{R}Y}{\mathbf{R}X + Y} \tag{3.22}$$

For $\alpha = 0$:

$$\begin{aligned} \frac{dX}{dY} &= \frac{\beta Y}{\beta X} \\ &= \frac{Y}{X} \end{aligned} \tag{3.23}$$

Similarly, in case 3) λ_1 and λ_2 are complex, and the system can be reduced to its canonical form:

$$\begin{cases} \frac{dX}{dt} &= \alpha X - \beta Y \\ \frac{dY}{dt} &= \beta X + \alpha Y \end{cases}$$

The trajectory can be obtained by solving function:

$$\frac{dX}{dY} = \frac{\alpha X - \beta Y}{\beta X + \alpha Y} \tag{3.24}$$

For $\alpha \neq 0$, divide both the numerator and the denominator on the right hand side in equation 3.24 by α,

$$\frac{dX}{dY} = \frac{X - \mathbf{R}Y}{\mathbf{R}X + Y} \tag{3.25}$$

For $\alpha = 0$:

$$\begin{aligned} \frac{dX}{dY} &= \frac{\beta Y}{\beta X} \\ &= \frac{Y}{X} \end{aligned} \tag{3.26}$$

The operation for these two cases does not change the equation but uses the parameter $\mathbf{R}$ instead to characterize its phase portrait.

For case 2) $\lambda_1 = \lambda_2$, there are two cases:

a) $\mu = 0$,

$$\frac{dX}{dY} = \frac{\alpha X}{\alpha Y} = \frac{X}{Y} \tag{3.27}$$

b) $\mu \neq 0$,

$$\frac{dX}{dY} = \frac{\alpha X}{\mu X + \alpha Y} \tag{3.28}$$

Because μ is arbitrary (*Lemma* 1), choose $\mu = \alpha$,

$$\begin{aligned} \frac{dX}{dY} &= \frac{\alpha X}{\mu X + \alpha Y} \\ &= \frac{\alpha X}{\alpha X + \alpha Y} \\ &= \frac{X}{X + Y} \end{aligned} \tag{3.29}$$

In this case, $\beta = 0$ and the value of α does not change the trajectory, and it can be characterized as $\mathbf{R} = 0$.

This completes the proof. $\diamond$

Corollary 2 *An alternative quantity to determine the local topological structure of a simple critical point is the angle* $\Theta = tan^{-1}(\frac{\beta}{\alpha})$.

Proof: From *theorem* 3, a local topological structure of a simple critical point can be characterize by $\frac{\beta}{\alpha}$ which is the same as $\tan\Theta$ (Figure 3.3). For $0 < \Theta < 2\pi$, there is a unique Θ corresponding to every $\mathbf{R}$.

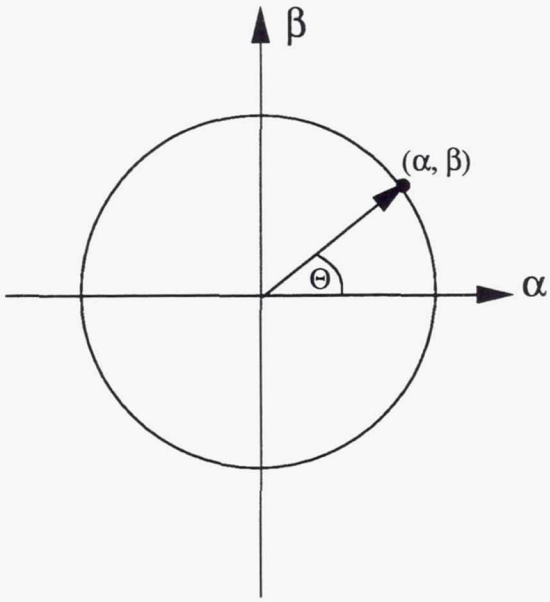


Figure 3.3: The point (α, β) and its equivalent angle Θ in $\alpha - \beta$ space.

This completes the proof. $\diamond$

The advantage of using Θ instead of $\mathbf{R}$ is that

- when $\alpha = 0, \beta = 1 \Longrightarrow \mathbf{R} = \infty, \Theta = \frac{\pi}{2}$ and
- when $\alpha = 0, \beta = -1 \Longrightarrow \mathbf{R} = -\infty, \Theta = \frac{3\pi}{2}$

Thus, all the values corresponding to various patterns are finite. The value Θ will be used for the rest of investigation.

Based on their α and β values, different dynamic systems have their corresponding points located in different regions of α-β space. From *theorem* 3, α and β themselves alone do not determine the trajectory of a dynamic system, the key value is their ratio. Therefore, a normalization procedure for α and β, namely,

$$\begin{aligned} \alpha' &= \frac{\alpha}{\sqrt{\alpha^2+\beta^2}} \\ \beta' &= \frac{\beta}{\sqrt{\alpha^2+\beta^2}} \end{aligned} \tag{3.30}$$

preserves the dynamic system. For the rest of the discussion, all the α's and β's have been normalized. The usefulness of normalization will be seen in calculating EMD (Section 3.4).

Corollary 3 *For normalized points in $\alpha - \beta$ space, the unit circle can be divided into (Figure 3.4:*

1) $\alpha = 1$, $\beta = 0$ and $\Theta = 0$, a repelling star or a repelling logarithmic node,

2) $\frac{\sqrt{2}}{2} < \alpha < 1$, $0 < \beta < \frac{\sqrt{2}}{2}$ and $0 < \Theta < \frac{1}{4}\pi$, a repelling node,

3) $-\frac{\sqrt{2}}{2} < \alpha < \frac{\sqrt{2}}{2}$, $\frac{\sqrt{2}}{2} < \beta < 1$ and $\frac{1}{4}\pi < \Theta < \frac{3}{4}\pi$,a saddle,

4) $-1 < \alpha < -\frac{\sqrt{2}}{2}$, $0 < \beta < \frac{\sqrt{2}}{2}$ and $\frac{3}{4}\pi < \Theta < \pi$, an attracting node,

5) $\alpha = -1$, $\beta = 0$ and $\Theta = \pi$, an attracting star or an attracting logarithmic node,

6) $-1 < \alpha < 0$, $-1 < \beta < 0$ and $\pi < \Theta < \frac{3}{2}\pi$, an attracting focus

7) $\alpha = 0$, $\beta = -1$ and $\Theta = \frac{3}{2}\pi$, a center,

8) $0 < \alpha < 1$, $-1 < \beta < 0$ and $\frac{3}{2}\pi < \Theta < 2\pi$, a repelling focus

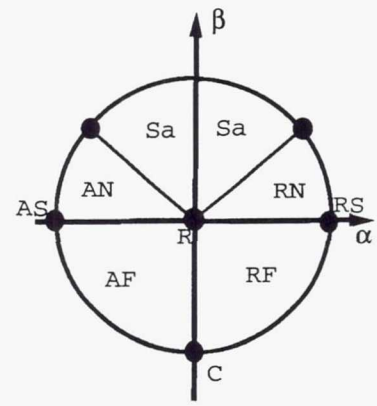


Figure 3.4: Basic patterns for critical points in α-β space;C for center, RN for node, AN for attracting node, RF for repelling focus, AF for attracting focus, St for star, Sa for Saddle and R for regular point

3.3.3 Simple critical points in α-β space

For a vector field with one simple critical point at $\vec{x}_0$, it can be approximated as

$$\vec{v} = (v_{xx}(x - x_0) + v_{xy}(y - y_0), v_{yx}(x - x_0) + v_{yy}(y - y_0))^2 \tag{3.31}$$

where

$$\begin{aligned} v_{xx} &= \left.\frac{\partial v_x}{\partial x}\right|_{\vec{x}_0} \quad v_{xy} = \left.\frac{\partial v_x}{\partial y}\right|_{\vec{x}_0} \\ v_{yx} &= \left.\frac{\partial v_y}{\partial x}\right|_{\vec{x}_0} \quad v_{yy} = \left.\frac{\partial v_y}{\partial y}\right|_{\vec{x}_0} \end{aligned} \tag{3.32}$$

Also, in Clifford algebra, this field can be represented by

$$\vec{v} = (az + b\bar{z} + c)e_1$$

where $a, b, c \in C$ and $z = x + iy$, $\bar{z} = x - iy$. Using the property:

$$e_1^2 = 1, \qquad e_1 e_2 = i = -e_2 e_1$$

[2]see Chapter 2

and let $a = a_1 + a_2 i$, $b = b_1 + b_2 i$ and $c = c_1 + c_2 i$ where $a_1, a_2, b_1, b_2, c_1 and c_2 \in R$.

$$\begin{aligned}
\vec{v} &= (a_1 + a_2 i)(x + iy) + (b_1 + b_2 i)(x - iy) + c_1 + c_2 i \\
&= (a_1 x - a_2 y + b_1 x + b_2 y + c_1) + (a_1 y + a_2 x - b_1 y + b_2 x + c_2) i \\
&= (a_1 x - a_2 y + b_1 x + b_2 y + c_1) e_1^2 + (a_1 y + a_2 x - b_1 y + b_2 x + c_2) e_1 e_2 \\
&= [(a_1 x - a_2 y + b_1 x + b_2 y + c_1) e_1) - (a_1 y + a_2 x - b_1 y + b_2 x + c_2) e_2] e_1
\end{aligned}$$

Therefore,

$$v_{xx} x + v_{xy} y - (v_{xx} x_0 + v_{xy} y_0) = (a_1 + b_1) x + (b_2 - a_2) y + c_1$$

$$v_{yx} x + v_{yy} - (v_{yx} x_0 + v_{yy} y_0) = -(a_2 + b_2) x + (b_1 - a_1) y + c_2$$

Equating all the x and y terms:

$$\begin{cases}
a_1 + b_1 &= v_{xx} \\
b_1 - a_1 &= v_{yy} \\
-(a_2 + b_2) &= v_{yx} \\
b_2 - a_2 &= v_{xy}
\end{cases} \tag{3.33}$$

and

$$c_1 = -(v_{xx} x_0 + v_{xy} y_0) \qquad c_2 = -(v_{yx} x_0 + v_{yy} y_0) \tag{3.34}$$

Solve equations 3.33,

$$\begin{aligned}
a_1 &= \tfrac{1}{2}(v_{xx} - v_{yy}) \qquad & b_1 &= \tfrac{1}{2}(v_{xx} + v_{yy}) \\
a_2 &= -\tfrac{1}{2}(v_{yx} + v_{xy}) \qquad & b_2 &= \tfrac{1}{2}(v_{xy} - v_{yx})
\end{aligned}$$

From equations 3.11 and 3.12, a pair of α, β values are readily given. After a normalization procedure 3.30, the corresponding point of the vector field 3.31 can be located on a unit circle in α-β space (Figure 3.4).

3.3.4 Regular points in α-β space

If a vector field on E does not have a singular point, then it is diffeomorphic to a constant field

$$\vec{v} = const \cdot e_1.$$

where $a = 0$ and $b = 0$. This gives $\alpha = 0$ and $\beta = 0$ which corresponds to the origin in the α-β space.

3.3.5 Multiple critical points in α-β space

An arbitrary vector field can be represented as a multiplication of linear fields (Section 3.2):

$$\begin{aligned} \vec{v}(r) &= E(z, \bar{z})e_1 \\ &= \prod_{k=1}^{n} (a_k z + b_k \bar{z} + c_k)e_1 \end{aligned} \tag{3.35}$$

where z_k is the unique zero of $a_k z + b_k \bar{z} + c_k$. Around each simple critical point $\vec{x}_k$, a vector field can be also expressed as $\vec{v}(\vec{x}) = \mathbf{J}_k(\vec{x} - \vec{x}_k)$ using its Jacobian $\mathbf{J}_k$. In section 3.3.3, a conversion from a vector field in regular space to a Clifford vector field is given. All the a_k's, b_k's and c_k's, therefore, can be computed separately at each critical point z_k. And a complex field with multiple critical points corresponds to multiple points $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)$ in α-β space. For example,

$$\vec{v} = \bar{z}^2 e_1$$

is a dipole which has two points both located at (1, 0) in α-β space (Figure 3.5).

3.3.6 The Growth of a Vector Field

A Clifford vector field

$$\vec{v}(r) = \prod_{k=1}^{n} (a_k z + b_k \bar{z} + c_k)e_1$$

Multiple Pages Missing from Available Version

where n is the total number of critical points in this field.

This energy here is a quantity that characterizes the critical points of a vector field. It is different from the physical energy. The concept "work" is used to measure the energy differences between two vector fields or the amount of energy used to transform one vector field into the other.

Definition 14 (Work) *For two vector fields with feature distributions*

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}$$

and

$$\{(\alpha'_1, \beta'_1), (\alpha'_2, \beta'_2), \ldots, (\alpha'_n, \beta'_n)\}.$$

The amount of work necessary for transforming one vector field into the other is defined as: $Work = \sqrt{\sum_{i=1}^{n}((\alpha_i - \alpha'_i)^2 + (\beta_i - \beta'_i)^2)}$.

Intuitively, given two feature distributions, one can be seen as a set of discrete point-objects with a certain amount of mass of earth spread in space, the other as a collection of holes in the same space. The work measures the least amount of energy needed to fill the holes with earth and is called the Earth Mover's Distance (EMD).

Computing the EMD is based on a solution to the old *transportation problem* from linear optimization. [72] This is a bipartite network flow problem which can be formalized as the following linear programming problem: Let I be a set of suppliers, J a set of consumers, c_{ij} the cost to ship a unit of supply from $i \in I$ to $j \in J$

$$c_{ij} = \sqrt{(\alpha_i - \alpha_j)^2 + (\beta_i^2 - \beta_j)^2}$$

and it is the same as the Euclidean distance $d_{ij} = ||\vec{v}_i - \vec{v}_j||$ in α-β space. A critical point either exists as a whole or does not exist, it can not be split. Therefore, the

flow f_{ij} can only be 0 or 1. We want to seek a set of f_{ij} that minimizes the overall cost:

$$EMD(\mathbf{x}, \mathbf{y}) = min \sum_{i \in I} \sum_{j \in J} c_{ij} f_{ij} \tag{3.40}$$

subject to the following constraints:

$$\begin{aligned} f_{ij} &\geq 0 \quad i \in I,\ j \in J && (3.41)\\ \sum_{i \in I} f_{ij} &= y_i, \quad j \in J && (3.42)\\ \sum_{j \in J} f_{ij} &\leq x_i, \quad i \in I && (3.43)\\ \sum_{j \in J} y_j &= \sum_{i \in I} x_i && (3.44) \end{aligned}$$

Where x_i is the total supply of supplier i and y_j is the total capacity of consumer j. Constraint (3.42) allows shipping of supplies from a supplier to a consumer and not vice versa. Constraint (3.43) forces the consumers to fill up all of their capacities and constraint (3.44) limits the supply that a supplier can send as a total amount. Constraint (3.44) is a feasibility condition that ensures that the total demand equals the total supply, in other words, the distributions have the same overall mass and the EMD is a true metric.

For a set of vector fields, they don't always have the same number of distributions. In order to satisfy constraint (3.44), we can create constant fields **1** with $\alpha = 0$ and $\beta = 0$ to make the supply equal to the demand without changing the vector fields. For example, the supplier field is

$$\vec{v}_1 = (2z + (1+i)\bar{z} + (-16+8i))((1+i)z + \bar{z} + (4-4i))(z + 2i\bar{z} + (-4+12i))e_1. \tag{3.45}$$

The critical points and their corresponding α, β values are listed in Table 3.1. There are three critical points in this field (Figure 3.7). And the consumer field is

$$\vec{v}_2 = (iz + (4+4i))(z + 2\bar{z} + (12+4i))(2z + (-8-8i))(\bar{z} + (4-4i))^2 e_1 \tag{3.46}$$

Multiple Pages Missing from Available Version

Order of critical point	Location	(α, β)
Simple	$(4, 4)$	$(0.5, 0.866)$
Simple	$(-4, -4)$	$(0.5774, 0.8165)$
Simple	$(-4, 4)$	$(0, -1.0)$

Table 3.1: Locations of three critical points and their corresponding α, β values; Order of critical point: the order of the lowest non-zero terms in the Taylor expansion of a critical point.

Order of critical point	Location	(α, β)
Simple	$(4, 4)$	$(0.0, 1.0)$
Simple	$(-4, -4)$	$(0.8944, 0.4472)$
Simple	$(-4, 4)$	$(1.0, 0.0)$
multiple (2nd order)	$(4, -4)$	$(0.0, 1.0)$ $(0.0, 1.0)$

Table 3.2: Locations of four critical points and their corresponding α, β values; Order of critical point: the order of the lowest non-zero terms in the Taylor expansion of a critical point.

This field has four critical points (Figure 3.8). Three of them are simple ones; however, the 4th one is a 2nd order point which means that there are two identical pairs of α, β values corresponding to this point. Therefore, there are five points in α-β space associated with these four points. The critical points and their corresponding α, β values are listed in Table 3.2. The supplier side has three points in the $\alpha - \beta$ space while the consumer side has five. Now let

$$\vec{v_1} = (2z + (1+i)\bar{z} + 1)((1+i)z + \bar{z} + i)(z + 2i\bar{z}) \cdot 1 \cdot 1e_1 \tag{3.47}$$

and the vector field remains unchanged. However, now we have two more regular points corresponding to 1 with $\alpha = 0$ and $\beta = 0$, and both the supplier and the consumer have 5 points in their feature distributions. All the conditions are satisfied, and we are ready to compute the EMD for these two fields and find out the dissimilarity between them.

3.5 Display of EMDs for a Large Set of Vector Fields

The above discussions are for comparison of a pair of vector fields. If there exist a large set of vector fields and we want to compare their topologies, it is necessary to display them in a more meaningful way than a 1D list sorted by their EMDs. Yossi Rubner et al. has used Multidimensional Scaling Method (MDS) [64, 65] to display a set of images on a 2D map. Given n objects in a high dimension, the MDS method computes a configuration of them in a lower dimension space such that the distance between every pair of objects in this low dimension space matches best to their real distance in the high dimension. Inspired by their work, we compute the EMDs between every pair of vector fields and position the vector fields on a 2D map such that the distances between the vector fields match their EMD values as close as possible.

3.6 Applications

Using a 2D map for displaying a set of vector fields can assist navigation in the space of vector fields. Computing the EMDs between pairs of selected vector fields and positioning them in a 2D map give us a better way to display the query results.

A simple example is shown in Figure 3.9. A list of simple critical points belonging to eight basic patterns are sorted by MDS to show the relationship (distances) between them. The result in Figure 3.9 shows that all the critical points fall into the right section in α-β space and are well matched to Figure 3.4. The sign of β controls whether the flow is circular or asymptotic. For $\beta > 0$, the vector field is asymptotic and the pattern is either a node or a saddle. For $\beta = 0$, it forms a star pattern. And for $\beta < 0$, it is circular and the pattern is a focus or a center. The sign of α controls

Multiple Pages Missing from Available Version

3.6.1 Unsteady Flow Past a 2D Cylinder

The feature comparison methods can also be used for comparing the topology between vector fields in different time steps. For example, Figure 3.13 shows an incompressible viscous flow past a 2D circular cylinder[3]. When the Reynolds number is higher than 40, the flow is unstable. Vortex shedding occurs behind the cylinder in the flow downstream. The alternatively shed vortex pattern is called a *Karman vortex street* [74]. Figure 3.12 shows the computations of EMDs between vector fields in different time steps with time step three. The topological patterns of the flow vary in a cyclic pattern due to the vortex shedding from the cylinder. The EMD for time step 3 and 18 is zero which means that the topology of the flow at these two time steps are indentical. This is shown in Figure 3.14. The only difference between these two fields is that the flow at time 18 is flipped upside down comparing to that of time 3. In Figure 3.12, the largest difference in one cycle is between time 18 and time 24 (or time 3 and time 8) where time 18 only has one center and time 24 has two centers and one saddle. This is shown in Figure 3.15.

3.7 Chapter Summary

Visualizing vector field topology is a very important subject in vector field visualization and has received much attention. However, to our knowledge, almost no work has been done on quantitative measurements for vector field comparisons. In this chapter, comparisons are based on feature distributions of vector fields. Critical points are key features of vector fields. Simple critical points are characterized by the eigenvalues of their Jacobian matrix. A new set of parameters α and β are introduced to replace the eigenvalues as criteria for critical point classification. The basic patterns of simple critical points can be classified into eight categories. A complex field can be

[3]details about the computation of this flow see Reference [73]

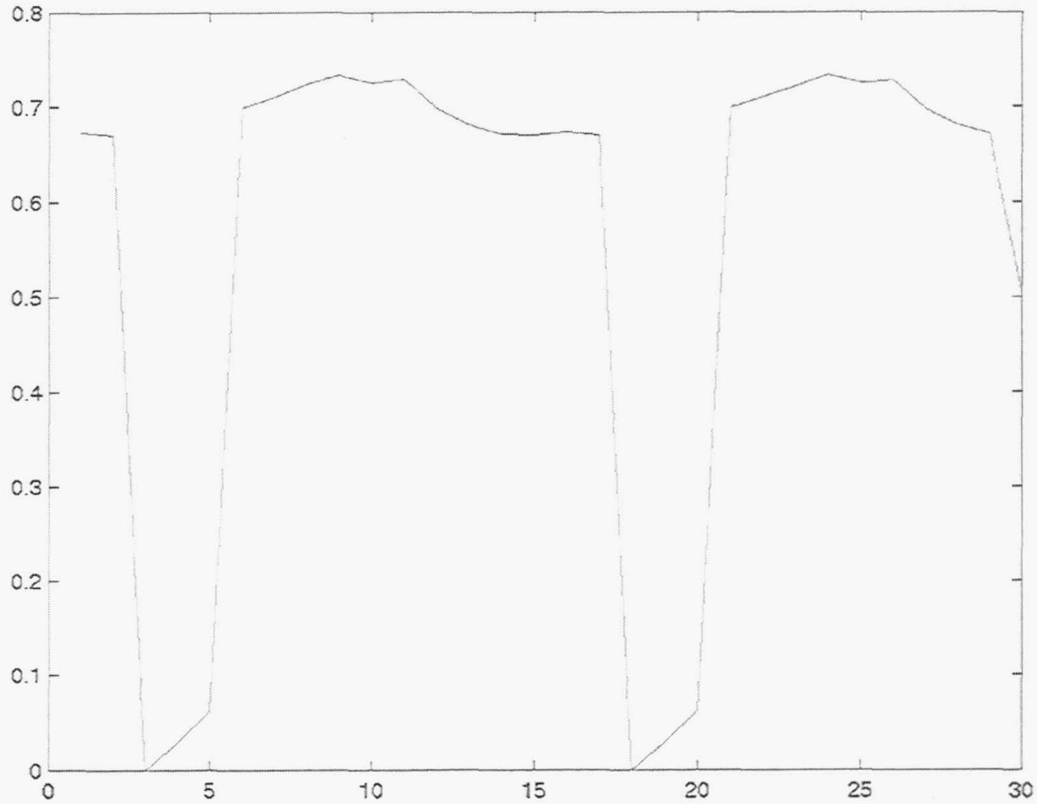


Figure 3.12: The EMDs for flows in different time steps with time step 3. The EMD for time step 3 and 18 is zero, which means that flow at these two time steps have the same topology.

Multiple Pages Missing from Available Version

Chapter 4

Topology of 3D Tensor Fields

Tensor field visualization is a very important area in multivariate multidimensional visualization. Due to their complex nature, tensor data sets usually contain huge volumes of information. Although they are seemingly random and chaotic, topology study shows that they are nicely structured and intrinsically correlated. Extracting the topological skeleton of a tensor field reveals its intrinsic geometric structure and captures the nature of the field. The topology of $2D$ tensor fields has been studied by Thierry Delmarcelle [38], this chapter intends to explore further into $3D$ space. The present research to our knowledge is original.

Degenerate points are the basic constituents of tensor field topology (Chapter 2). A thorough study of the behavior of a tensor field in a close neighborhood of its degenerate points can lead to a simple topological skeleton that connects these points. Because the integral lines of eigenvectors (or hyperstreamlines) in a tensor field never cross each other, except at degenerate points, one can reconstruct the whole tensor field based only on a small fraction of the data set. In this manner, we are able to compress the data substantially with no information loss, but it should be noted that this compression is different from the usual approach used in computer science. Also, by displaying only the topological skeleton, we can avoid the visual clutter yet still

reveal the essential features of the field.

Studies on first-order (vector) and second-order tensor fields show that their topologies have striking similarities. One wonders why a point with the same eigenvalues in a tensor field behaves in a similar fashion to a singular point (a point where the magnitude vanishes) in a vector field. It will be explained in this Chapter.

Every tensor field can be decomposed into a deviator and a spherical part (definitions to follow). The spherical part is an isotropic tensor and therefore remains invariant to a coordinate system transformation. As a result, there is no particular need to study this part of the tensor (here we refer to it as "the isotropic tensor"). It will be shown that the deviator of a tensor is parallel to the tensor itself. Therefore, their respective eigenvector fields are identical. Furthermore, the locations of the respective degenerate points are also identical. This, in turn, means that the topology of a tensor field is identical to the topology of its deviator.

In order to study the topology of degenerate points, it is critical to locate these points first. In this chapter, a method for locating isolated triple degenerate points will be presented. Cases of degenerate points of double degeneracy are more complicated, they may appear along lines or surfaces. In this chapter, the concept of a "control function" of a **deviator** is proposed. This function determines the existence of degenerate points and assists in defining the lines and surfaces along which the degenerate points lie. Following the study of the tensor field in the neighborhood of degenerate points, one can display a full representation of the 3-D tensor field.

4.1 Decomposition of a tensor field

A tensor field in the real physical world is often very complex, therefore reducing it into a simpler form becomes very appealing. In this section, the tensor is decomposed into a "deviator" and an "isotropic tensor" at each point in the tensor field.

Definition 15 (Deviator) *A tensor is a deviator* $\boldsymbol{D}$ *iff it is trace free, i.e.,* $Trace\,(\boldsymbol{D}) = 0$.

Definition 16 (Isotropic Tensor) *A tensor is isotropic iff* $U_{ij} = \nu\delta_{ij}$, *where* ν *is a stretch factor.*

Any given tensor $\mathbf{T}$, can be decomposed into[1]:

$$\begin{aligned}
&\begin{pmatrix} T_{11}(x,y,z) & T_{12}(x,y,z) & T_{13}(x,y,z) \\ T_{21}(x,y,z) & T_{22}(x,y,z) & T_{23}(x,y,z) \\ T_{31}(x,y,z) & T_{32}(x,y,z) & T_{33}(x,y,z) \end{pmatrix} \\
&= \begin{pmatrix} D_{11}(x,y,z) & D_{12}(x,y,z) & D_{13}(x,y,z) \\ D_{21}(x,y,z) & D_{22}(x,y,z) & D_{23}(x,y,z) \\ D_{31}(x,y,z) & D_{32}(x,y,z) & D_{33}(x,y,z) \end{pmatrix} \\
&+ \begin{pmatrix} U_{11}(x,y,z) & 0 & 0 \\ 0 & U_{22}(x,y,z) & 0 \\ 0 & 0 & U_{33}(x,y,z) \end{pmatrix}
\end{aligned} \tag{4.1}$$

where

$$D_{ii} = T_{ii} - \frac{1}{3}\sum_{j=1}^{3} T_{jj}$$

and

$$U_{ii} = \frac{1}{3}\sum_{j=1}^{3} T_{jj}.$$

Therefore,

$$D_{11} + D_{22} + D_{33} = 0$$

and

$$U_{11} = U_{22} = U_{33}$$

[1]the analysis and examples in the rest of the chapter are all for 3-D space, but the theory holds for 2-D space as well

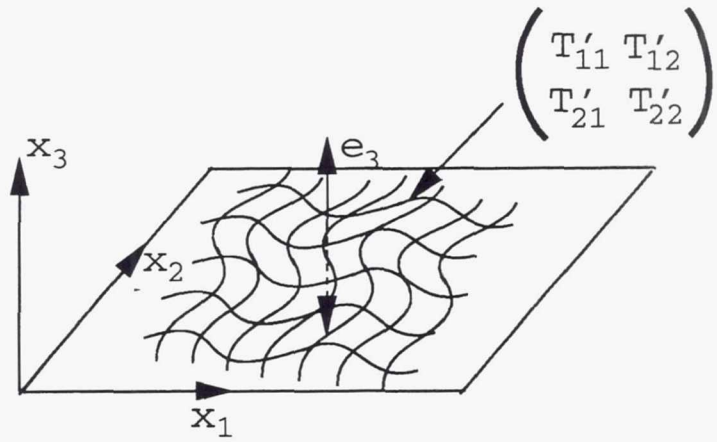


Figure 4.1: Eigenvector e_3 is aligned to x_3, a local 2D tensor on the plane.

at any point in the field. Here the deviator and the isotropic tensor are denoted by **D** and by **U**.

A tensor can also have a "local 2-D deviator". Any given tensor can be transformed into a space so that at least one eigenvector, for example $\vec{e_3}$, is along one of the axes. In the plane perpendicular to $\vec{e_3}$, the tensor is locally 2-D (Figure 4.1). It takes the form:

$$\begin{pmatrix} T'_{11} & T'_{12} & 0 \\ T'_{12} & T'_{22} & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

The locally 2-D tensor can be decomposed into a deviator

$$D' = \begin{pmatrix} D'_{11} & D'_{12} \\ D'_{12} & D'_{22} \end{pmatrix}$$

and an isotropic tensor

$$U' = \begin{pmatrix} U'_{11} & 0 \\ 0 & U'_{22} \end{pmatrix}.$$

Multiple Pages Missing from Available Version

the eigenvalues at these points are equal to zero. This means that these points are stress free; a fact that can be verified by an examination of the stress equations. We have therefore acquired physical insight into the stress tensor field just by examining a basic topological feature, a point of triple degeneracy. Also, after searching through the whole field, we found that there are a continuous line of double degenerate points inside the solid. The stress tensor has a singular "local deviator" along this line.

Figure 4.2 shows the minor eigenvector field (most compressive force). The arrows represent the two forces, the two balls indicate the location of singular points. The hyperstreamlines on the top surface are the separatrices of the singular points which describe the topological structure in their vicinity. Inside the solid, the hyperstreamlines form trisectors of double degenerate points which indicate the pattern of topological structure in the vicinity of their singular local 2-D deviators. Colors of the hyperstreamlines encode the magnitudes of minor eigenvalues. We also varied the strength of the two forces . It turns out that the stress free points move along a circle as the force ratio is varied. Video clip 1 shows the motion of a double degenerate point inside the solid with variation of the two forces; Video clip 2 shows the motion of two triple degenerate points on the top surface(Figure 4.2) with variation of two forces; Red balls indicates the degenerate points.

4.2 Physical meaning of a deviator and an isotropic tensor

4.2.1 Isotropic Tensor

Based on section 4.1, one can see that the tensor **U** is an identity tensor multiplied by U_{ii}. Therefore, at any given location in the field, it behaves the same in every direction, in other words, it is an isotropic tensor. Since it is isotropic throughout the

whole field, it is of no particular physical interests to the following study.

4.2.2 Deviator

The deviator, in contrast to the isotropic tensor, has a different behavior in all 3 principal directions except at a singular point where all of its components are zero. It is indeed the deviator that represents the deviations from the originally isotropic field and makes the entire tensor field so complex and diverse. It is then reasonable to think of a real tensor field as a **deviator** superimposed onto an **isotropic tensor**. By subtracting the contribution of the **isotropic tensor** from the tensor field, the **deviator** becomes dominant and it enables us to clearly show the topology and the fluctuations of the field without the disturbance of the sometimes dominant isotropic contribution. One can think of this as a way to reduce the effect of the constant background, while making the variations from constancy more pronounced. The major focus is then on the **deviator**, as all the following sections are dedicated to the analysis of the topology of the **deviator**.

4.2.3 Stress Tensor *vs.* Viscous Stress Tensor—An Example

This section shows an example of different effects of an isotropic tensor and a deviator in a flow field. The flow is incident on the front of the hemisphere cylinder and proceeds to the back of the body at a skewed angle. The viscous stress tensor in the field is expressed as:

$$\sigma_{ij} = \mu(\frac{\partial \nu_i}{\partial x_j} + \frac{\partial \nu_j}{\partial x_i})$$

where $i, j = 1, 2, 3$. And the stress tensor is:

$$\sigma'_{ij} = -P\delta_{ij} + \sigma_{ij}$$

where P is the pressure. The only difference between these two tensors is that the stress field has a uniform pressure component. Pressure in a stress tensor comes

Multiple Pages Missing from Available Version

from outside forces acting on the medium and it can be disturbingly large. In Figure 4.3, hyperstreamlines (Chapter 1) are integrated along major eigenvectors, and the medium and minor eigenvectors residing in the cross sections. The size of the tube is determined by the values of the medium and minor eigenvalues. Because the pressure component in this field is rather large, some small variations from the viscous part won't make an impact on the appearances of the tubes and they all appear circular. This makes it difficult to see the actual effect from the flow itself. However, in Figure 4.4, the isotropic contribution is subtracted from the tensor. Therefore, pressure no longer has any effect and the resultant tensor is a deviator. The hyperstreamlines are now all anisotropic, which means that the viscous stresses are very different in the directions of the medium and minor eigenvectors. From the comparison of these two figures, we can see that a deviator truly reflects the nature of a tensor field.

4.3 Degenerate Points in a Deviator

Because the trace of a **deviator** is zero, degenerate points in a deviator take on some special features.

Definition 17 (Singular Point) *A singular point in a tensor field is a point where all eigenvalues of a tensor vanish, in mathematical representation, it is a zero matrix.*

For a deviator (of a symmetric tensor),

$$\begin{pmatrix} D_{11}(x_0) & D_{12}(x_0) & D_{13}(x_0) \\ D_{12}(x_0) & D_{22}(x_0) & D_{23}(x_0) \\ D_{13}(x_0) & D_{23}(x_0) & D_{33}(x_0) \end{pmatrix}$$

a triple degenerate point satisfies the conditions:

$$\begin{cases} D_{11} - D_{22} &= 0 \\ D_{22} - D_{33} &= 0 \\ D_{12} &= 0 \\ D_{13} &= 0 \\ D_{23} &= 0 \end{cases}$$

From the first two conditions, we get

$$D_{11} = D_{22} = D_{33},$$

and also

$$D_{11} + D_{22} + D_{33} = 0,$$

therefore,

$$D_{11} = 0 \quad D_{22} = 0 \quad D_{33} = 0$$

So, all the components of a triple degenerate point are zero and this is a singular point similar to a critical point in a first-order tensor field (vector field) where all the components vanish. A double degenerate point in a local deviator

$$\begin{pmatrix} D'_{11} & D'_{12} & 0 \\ D'_{12} & D'_{22} & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

also brings

$$D'_{11} = 0 \quad D'_{22} = 0 \quad D'_{12} = 0$$

This corresponds to a point where only two components vanish $(0, 0, v_3)$ in a vector field. However, both of them are dependent on the coordinates. These features of triple and double degenerate points shed some light on the similarities of topological properties between the first- and second-order tensor fields.

4.4 The Existence Conditions of Hyperstreamlines in Q-R Space

The eigenvalues of a 3-D tensor **T** determine the existence of its trajectories and can be obtained by investigating its characteristic equation.

$$\begin{aligned} A(\lambda) &= \begin{vmatrix} \lambda - T_{11} & -T_{12} & -T_{13} \\ -T_{12} & \lambda - T_{22} & -T_{23} \\ -T_{13} & -T_{23} & \lambda - T_{33} \end{vmatrix} \\ &= \lambda^3 + P\lambda^2 + Q\lambda + R \end{aligned} \tag{4.3}$$

where:

$$P = T_{11} + T_{22} + T_{33} \tag{4.4}$$

$$Q = \begin{vmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{13} \\ T_{13} & T_{33} \end{vmatrix} + \begin{vmatrix} T_{22} & T_{23} \\ T_{23} & T_{33} \end{vmatrix} \tag{4.5}$$

$$R = \begin{vmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{vmatrix} \tag{4.6}$$

The coefficients P, Q and R are all tensor invariants.

Theorem 5 *In P-Q-R space, hyperstreamlines only exist between surfaces S_1 and S_2, which are, respectively, given by:*

$$\frac{2P^3 + 9PQ + 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.7}$$

and

$$\frac{2P^3 + 9PQ - 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.8}$$

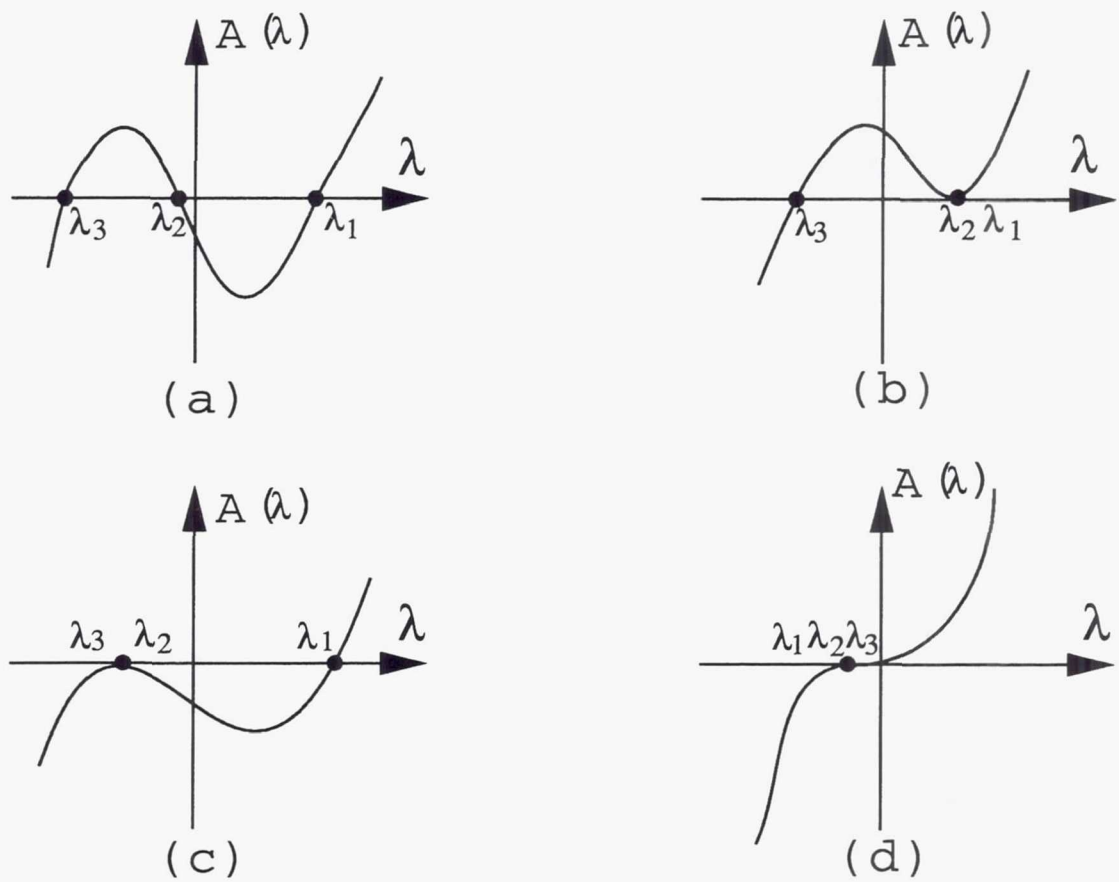


Figure 4.5: (a) $A(\lambda)$ is a general cubic polynomial with three real distinct roots; (b) $A(\lambda)$ has one double root and one single root: $\lambda_1 = \lambda_2 > \lambda_3$ (c) $A(\lambda)$ has one double root and one single root: $\lambda_1 > \lambda_2 = \lambda_3$; (d) $A(\lambda)$ has one triple root: $\lambda_1 = \lambda_2 = \lambda_3$.

Proof: Since the tensor **T** is real and symmetric, all eigenvalues must be real. Its characteristic function $A(\lambda)$ is a cubic polynomial, therefore, it must have three real solutions (Figure 4.5). When $A(\lambda)$ has double roots, it satisfies the conditions:

$$\begin{cases} A(\lambda) = \lambda^3 + P\lambda^2 + Q\lambda + R = 0 \\ \frac{dA(\lambda)}{d\lambda} = 3\lambda^2 + 2P\lambda + Q = 0 \end{cases} \tag{4.9}$$

Solving equations 4.9 gives two condition equations

$$E_1 = \frac{2P^3 + 9PQ + 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.10}$$

and

$$E_2 = \frac{2P^3 + 9PQ - 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.11}$$

corresponding to $\lambda_1 = \lambda_2 > \lambda_3$ and $\lambda_1 > \lambda_2 = \lambda_3$, respectively. Therefore,

- when $E_1 < 0$ and $E_2 > 0$, there are three real distinct roots (Figure 4.5 (a)),
- when $E_1 = 0$ and $E_2 > 0$, one double root and one single root exist (Figure 4.5 (b)),
- when $E_1 > 0$ and $E_2 = 0$, one double root and one single root (Figure 4.5 (c)) exist and
- when $E_1 = E_2$, one triple root (Figure 4.5 (d)) exist.

From Figure 4.5, obviously if $E_1 > 0$ or $E_2 < 0$ then $A(\lambda)$ will have only one real root and a pair of complex conjugate roots. Therefore, the conditions for all λ's to be real or hyperstreamlines to possibly exist are that $E_1 \leq 0$ and $E_2 \geq 0$. In P-Q-R space, this is equivalent to saying that hyperstreamlines only exist in between surfaces:

$$E_1 = \frac{2P^3 + 9PQ + 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.12}$$

and

$$E_2 = \frac{2P^3 + 9PQ - 2\left(P^2 - 3Q\right)^{3/2}}{27} + R = 0 \tag{4.13}$$

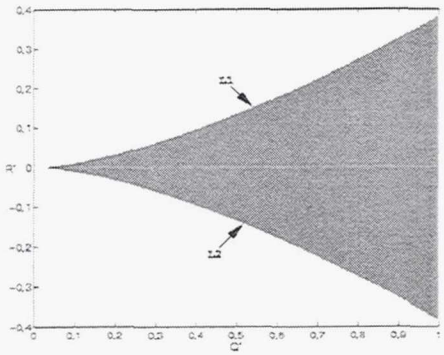


Figure 4.6: Q'-R' plane

This completes the proof. ◇

On surface S_1, $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0)$; and on surface S_2, $\lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0)$. Triple degeneracy $\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0)$ occurs at points where S_1 and S_2 meet, i.e. $R = \frac{P^3}{27}$, $Q = \frac{P^2}{3}$.

Since the tensor and its deviator have the same topology, it is sufficient to examine only the deviator $\mathbf{D}$. By definition,

$$D_{11} + D_{22} + D_{33} = 0,$$

and therefore, $P = 0$. The coefficient Q can also be presented as:

$$Q' = -\frac{1}{2}(D_{11}^2 + D_{22}^2 + D_{33}^2 + 2D_{12}^2 + 2D_{13}^2 + 2D_{23}^2) \tag{4.14}$$

The characteristic equation now becomes:

$$\lambda^3 + Q'\lambda + R' = 0 \tag{4.15}$$

And the P-Q-R space reduces to the Q'-R' plane.

Corollary 4 *In $Q' - R'$ space, the hyperstreamlines exist between curves L_1 and L_2, which are, respectively, given by:*

$$\frac{R'}{2} = \left(\frac{-Q'}{3}\right)^{\frac{3}{2}} \tag{4.16}$$

and

$$\frac{R'}{2} = -\left(\frac{-Q'}{3}\right)^{\frac{3}{2}} \tag{4.17}$$

Proof: From theorem 5, the hyperstreamlines only exist between surfaces: S_1 and S_2. Now substitute $P = 0$ into equations 4.10 and 4.11, S_1 and S_2 reduce to curve L_1 and L_2 on the $Q' - R'$ plane(Figure 4.6), given respectively by

$$\frac{R'}{2} = \left(\frac{-Q'}{3}\right)^{\frac{3}{2}} \tag{4.18}$$

and

$$\frac{R'}{2} = -\left(\frac{-Q'}{3}\right)^{\frac{3}{2}} \tag{4.19}$$

This completes the proof. ◇

Double degeneracy occurs on the curves L_1 and L_2 and triple degeneracy occurs where L_1 and L_2 meet, i.e. $Q' = R' = 0$.

4.5 Control functions of the singular points

From section 4.4, the characteristic equation of a deviator takes the form:

$$\lambda^3 + Q\lambda + R = 0.$$

The standard form for the roots of this cubic equation is:

$$\lambda_1 = \sqrt[3]{-\frac{R}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{R}{2} - \sqrt{\Delta}},$$

$$\lambda_2 = \omega\sqrt[3]{-\frac{R}{2} + \sqrt{\Delta}} + \omega^2\sqrt[3]{-\frac{R}{2} - \sqrt{\Delta}},$$

$$\lambda_3 = \omega\sqrt[3]{-\frac{R}{2} - \sqrt{\Delta}} + \omega^2\sqrt[3]{-\frac{R}{2} + \sqrt{\Delta}}$$

Where

$$\Delta = \left(\frac{R}{2}\right)^2 + \left(\frac{Q}{3}\right)^3$$

and

$$\omega = \frac{-1 + i\sqrt[3]{3}}{2}.$$

This equation has three distinct real roots when $\Delta < 0$ and multiple real roots when $\Delta = 0$. This study deals with symmetric tensors, and therefore the eigenvalues are always real. This, in turn, means that for all regular points in the tensor field Δ has to be less than 0. The quantity $\Delta = 0$ only at a degenerate point, where the characteristic equation gives multiple roots.

- For double roots, $(\frac{R}{2})^2 = -(\frac{Q}{3})^3 \neq 0$, it is a double degenerate point,
- For triple roots, $Q = R = 0$, it is a triple degenerate point.

The existence conditions derived in section 4.4 can also be obtained from $\Delta(x, y, z) = 0$, but from a different point of view. Function $\Delta(x, y, z)$ controls the occurrences of degeneracy of a deviator, thus it is called a "Control Function".

4.6 The nature of a degenerate point

Theorem 6 *A tensor is of triple degeneracy iff its deviator is singular.*

Proof: If a tensor $\mathbf{T}$ is of triple degeneracy, then $\mathbf{T}$ and its isotropic tensor $\mathbf{U}$ are identical. Therefore, its deviator $\mathbf{D} = \mathbf{T} - \mathbf{U} = \mathbf{0}$, or $\mathbf{D}$ is singular; on the other hand, if $\mathbf{D}$ is singular, then $\mathbf{T} = \mathbf{D} + \mathbf{U} = \mathbf{T}$.

This completes the proof. $\diamond$

Theorem 7 *A tensor is of double degeneracy iff its deviator is nonsingular and one of its local 2-D deviators is singular.*

Proof: If a tensor **T** is of double degeneracy, let its eigenvalues be λ_1, λ_2 and λ_3, and also $\lambda_1 = \lambda_2$, then it has one distinct eigen direction $\vec{e_3}$ associated with λ_3. We rotate the coordinate system so that one axis is along $\vec{e_3}$. In the transverse plane, the tensor **T'** is locally 2-D and degenerate, therefore, it is identical to its isotropic tensor **U'** and its deviator **D'**(also a local 2-D deviator of **T**) is singular. However, if the deviator **D** of **T** is singular then from Theorem 6, **T** has to be of triple degeneracy. Since **T** is only of double degeneracy, **D** must be nonsingular.

On the other hand, if one of its local 2-D deviators **D'** is singular, then the local 2-D tensor **T'** = **U'** and **T** is degenerate. Since the deviator **D** is nonsingular, from Theorem 6, **T** is not of triple degeneracy. Therefore, **T** must be of double degeneracy.

This completes the proof. ◇

Singular points are the basic constituents of both vector fields and deviators. A singular deviator can be related to a vector with three zero components. It is natural to see the resemblance between them in topology. Since a tensor and it's deviator have the same set of eigenvectors and their degenerate points occur at the same place, they share the same topological structure. This is why the topological structure of a first-order tensor(vector) and a second-order tensor field have such a striking similarity. In fact, although a second-order field is more complex than a first-order field, because the eigenvector fields of a second-order tensor field has sign indeterminacy, it usually has an even simpler structure.

4.7 Representation of singularities in deviators

The existence and location of degenerate points in a tensor field is determined by the Control Function. Analysis of the Control Function behavior in the neighborhood of a degenerate point also contains information regarding the distribution of singular

points in a tensor field especially in the case that singular points appear along a continuous line or surface.

The control function $\Delta(x, y, z)$ has maxima at a degenerate point $\vec{x_0}$,

$$\begin{cases} \Delta(\vec{x_0}) = 0 \\ \frac{d\Delta(\vec{x_0})}{d\vec{x}} = 0 \end{cases} \tag{4.20}$$

In the vicinity of this point, the control function can be expressed as:

$$\begin{aligned} \Delta(\vec{x}) &= \Delta(\vec{x_0}) + \sum_{i=1}^{3} \frac{\partial \Delta(\vec{x_0})}{\partial x_i}(x_i - x_{0i}) \\ &+ \frac{1}{2!}\left(\sum_{i=1}^{3}(x_i - x_{i0})\frac{\partial}{\partial x_i}\right)^2 \Delta(\vec{x_0}) + \cdots \end{aligned} \tag{4.21}$$

If a singular point is located on a continuous line or surface of singular points then the condition: $\Delta(\vec{x}) = 0$ is satisfied along the line or on the surface, respectively. Substituting Equation 4.20 into Equation 4.21, results in the following:

$$\begin{aligned} \Delta(\vec{x}) &= \frac{1}{2!}\left(\sum_{i=1}^{3}(x_i - x_{i0})\frac{\partial}{\partial x_i}\right)^2 \Delta(\vec{x_0}) \\ &+ \frac{1}{3!}\left(\sum_{i=1}^{3}(x_i - x_{i0})\frac{\partial}{\partial x_i}\right)^3 \Delta(\vec{x_0}) + \cdots \\ &= 0 \end{aligned} \tag{4.22}$$

In Equation 4.22, the control function $\Delta(\vec{x})$ is expanded to the *nth* order so that at least one of its *nth* partial derivatives at $\vec{x_0}$ is nonzero [76] [77]. Rearranging the function according to the order of variable $(x - x_0)$(here $\vec{x} - \vec{x_0}$ is written as $x - x_0$, $y - y_0$ and $z - z_0$), results in an implicit equation in 3-D space:

$$\begin{aligned} 0 &= f_{n,0,0}(x - x_0)^n + f_{n-1,1,0}(x - x_0)^{n-1}(y - y_0) + \\ &\quad f_{n-1,0,1}(x - x_0)^{n-1}(z - z_0) + \cdots + \\ &\quad f_{0,n,0}(y - y_0)^n + f_{0,0,n}(z - z_0)^n \end{aligned} \tag{4.23}$$

Depending on the values of the coefficients $f_{i,j,k}$'s, this equation is a representation of an object in 3-D space; surfaces, lines and isolated points are all possibilities [76]. The following is an example of an analytical tensor:

$$\mathbf{T} = \begin{pmatrix} x+y-\frac{1}{2} & x-y & 0 \\ x-y & -(x+y)+\frac{3}{2} & 0 \\ 0 & 0 & z \end{pmatrix} \tag{4.24}$$

By computing the trace and decomposing the tensor into its deviator and the isotropic tensor:

$$\mathbf{D} = \begin{pmatrix} x+y-\frac{5}{6}-\frac{z}{3} & x-y & 0 \\ x-y & -(x+y)+\frac{7}{6}-\frac{z}{3} & 0 \\ 0 & 0 & -\frac{1}{3}+\frac{2z}{3} \end{pmatrix} \tag{4.25}$$

and

$$\mathbf{U} = \begin{pmatrix} \frac{1}{3}(z+1) & 0 & 0 \\ 0 & \frac{1}{3}(z+1) & 0 \\ 0 & 0 & \frac{1}{3}(z+1) \end{pmatrix} \tag{4.26}$$

One can find the coefficients Q and R(Equation 4.6 and Equation 4.14) of the characteristic equation,

$$\begin{aligned} Q &= -\frac{1}{2}\left(D_{11}^2 + D_{22}^2 + D_{33}^2 + 2D_{12}^2 + 2D_{13}^2 + 2D_{23}^2\right) \\ &= -4\left(x-\frac{1}{2}\right)^2 - 4\left(y-\frac{1}{2}\right)^2 - \frac{2}{3}\left(z-\frac{1}{2}\right)^2 \end{aligned} \tag{4.27}$$

$$\begin{aligned} R &= \begin{vmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{vmatrix} \\ &= -\frac{2}{3}\left[2\left(x-\frac{1}{2}\right)^2 + 2\left(y-\frac{1}{2}\right)^2 - \frac{1}{9}\left(z-\frac{1}{2}\right)^2\right] \\ &\quad \left(z-\frac{1}{2}\right) \end{aligned} \tag{4.28}$$

And then find the degenerate points as follows:

- Triple Degeneracy(triple roots): $Q = R = 0$

  Result: an isolated point: $(x, y, z) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$.

- Double Degeneracy(double roots): $\left(\frac{R}{2}\right)^2 = -\left(\frac{Q}{3}\right)^3 \neq 0$

Results:

- a line: $x = \frac{1}{2}$, $y = \frac{1}{2}$
- a surface: $\left(x - \frac{1}{2}\right)^2 + \left(x - \frac{1}{2}\right)^2 = \frac{1}{2}\left(z - \frac{1}{2}\right)^2$

The line and surface described above actually include both cases of double degeneracy:

- $\lambda_1 = \lambda_2 > \lambda_3$: $x = \frac{1}{2}$, $y = \frac{1}{2}$ and $z < \frac{1}{2}$ for the line; $\sqrt{(x - \frac{1}{2})^2 + (x - \frac{1}{2})^2} = \frac{\sqrt{2}}{2}(z - \frac{1}{2})$ for the surface.
- $\lambda_1 > \lambda_2 = \lambda_3$: $x = \frac{1}{2}$, $y = \frac{1}{2}$ and $z > \frac{1}{2}$ for the line; $\sqrt{(x - \frac{1}{2})^2 + (x - \frac{1}{2})^2} = -\frac{\sqrt{2}}{2}(z - \frac{1}{2})$ for the surface.

Figure 4.7 shows the resulting degeneracy: the small red ball is the location of a triple degenerate point; the line and the surface in the top figure is for $\lambda_1 = \lambda_2 > \lambda_3$ case and the line and the surface in the bottom figure is for $\lambda_1 > \lambda_2 = \lambda_3$.

4.8 Topological Structure of a Singular Point in a 3D deviator

4.8.1 The Separating Surface of a Singular Point

For second order tensor fields, in most cases, the eigenvector fields in the vicinity of a degenerate point can be described in terms of three types of angular sectors:

Multiple Pages Missing from Available Version

in the vicinity of the points of triple degeneracy. They also indicate that a locus of points of double degeneracy ($\lambda_2 = \lambda_3$) connects the points of triple degeneracy. This is evident from the two trisector points that lie in the symmetry planes just below the points of triple degeneracy. The existence of the line of double degeneracy is further verified by noting the two points of double degeneracy in the skeleton of the medium hyperstreamlines (Figure 4.13).

4.9 Applications

The following section contains examples of application using the methods discussed in this chapter applied to problems of scientific interest. We use both texture mapping and hyperstreamlines for display[2].

Deformation Tensor

In the case of incompressible flow the deformation tensor, defined as

$$\mathbf{Def} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}$$

has a zero isotropic part and therefore is equal to its deviator.

For compressible flow, the deformation tensor is composed of a deviator superimposed on a non-zero isotropic tensor which represents the rate of expansion. Therefore, a deviator describes the topological structure for both incompressible and compressible flows.

For a rotational flow, inside the vortex core, flow is pure rotational. Assume that the flow advances in the z-direction and rotates around the z-axis while the velocity within the vortex core area is

$$(-\omega y, \omega x, 0)$$

[2]For dipole techniques, see Chapter 1

where ω is the angular velocity. By definition, the deformation tensor $\mathbf{Def}(r < R)$ becomes singular; outside the vortex core, the velocity is

$$\frac{\Gamma}{r^2}(-y, x, 0)$$

and its deformation tensor $\mathbf{Def}(r > R)$ is:

$$\frac{2\Gamma}{r^2}\begin{pmatrix} xy & -x^2+y^2 & 0 \\ -x^2+y^2 & -xy & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Here r is the distance from a point to the center of the vortex core and R is the radius of the vortex core. It is virtually a 2-D tensor with major and minor eigenvalues having equal magnitude but opposite sign and the medium eigenvalue remains zero. The deformation tensor is discontinuous at $r = R$. The angles of separatrices are calculated by using the tensor in the neighborhood of the vortex core $\mathbf{Def}(r > R)$ [38], and it turns out that there is no real solution for the angles. This indicates that the major and minor eigenvector fields are a pair of loci in the transverse plane while the medium eigenvector follows the direction of the vortex core. Studies of the alignment between vorticity and eigenvectors of the strain-rate (deformation) tensor in numerical solutions of Navier-Stokes turbulence have shown that the two principal strains with the largest absolute values (major and minor eigenvectors) lie in the equatorial plane, and the vorticity is automatically aligned to the intermediate eigenvector.[3]

Figure 4.14 (top) and Figure 4.14 (bottom) show the texture of a flow past a wingtip for a major eigenvector field and a minor eigenvector field, respectively. These two eigenvectors remain in the transverse plane perpendicular to the vortex core. Images are taken from a slice of the transverse plane along the vortex core, and the two eigenvector fields form two loci as we might expect. Color encodes the magnitude of their associated eigenvalues.

[3]for detailed information, please see [27] [26].

Multiple Pages Missing from Available Version

suggested that the model can be improved by taking into account the variation in swirling speed.

4.10 Chapter Summary

Tensor analysis is a very challenging task due to its complexity. Simplification is a natual choice. This chapter introduces a decomposition procedure to break a tensor field into a deviator and an isotropic tensor. A deviator carries the essential information about the tensor field, and describes the same topological structure. An isotropic tensor provides extra information like pressure in a stress field or expansion of a compressible flow in a deformation tensor, it is uniform at every location in the field yet can be dominant compared to the deviator. In order to understand the nature of a tensor field, we primarily study its deviator.

Tensor invariants are very important quantities in tensor analysis. The characteristic equation of a general tensor has three invariants P, Q and R while a deviator only has two Q' and R'. These invariants determine that hyperstreamlines only exist in between two surfaces in $P - Q - R$ space and two curves in $Q' - R'$ space, respectively.

The degeneracy of a deviator is also its singularity, which explains the similarity between vector and second-order tensor fields. Because of the extreme importance of degenerate points to tensor analysis, the conditions for the existence of degenerate points are presented here. A control function is derived from these conditions which enables us to predict the shape of this point set in a 3-D tensor field. The study of a control function sheds light on the representation of singularities in a deviator as well as in the tensor field itself.

The singularities in a tensor field often have a very important physical meaning. For a deformation tensor, singularities only occur either at a vortex core or in a

constant flow; for a stress tensor in solids, it is a stress free point. After being decomposed into a deviator and an isotropic tensor, a stress tensor in a viscous flow with high pressure reveals the anisotropy of the flow. We applied our methods to several physical problems and the results are very encouraging.

Chapter 5

Conclusions and Future Research

Vector and tensor fields are multidimensional multivariate data sets and are very difficult to comprehend. The objective of visualization is to simplify the analysis yet still capture the key tensor properties. This dissertation discusses the advantages and methods for visualizing and analyzing vector and tensor fields using topology.

Because of its close connections with differential equations, combinatorial topology is the desired tool for studying vector and tensor fields. It deals with geometry, thus is able to find the intrinsic properties—critical and degenerate points and their corresponding separatrices that define the frame work of the fields (topological skeletons). In this dissertation, research efforts on visualization developments for vector and tensor fields are based on topology analysis.

5.1 Contributions

5.1.1 Feature Comparisons for Vector Fields

Much research has been done on vector field visualization. However, virtually none is on quantitative comparisons. In this dissertation, we define "energy" as a quantity to

describe the feature distribution of a vector field. Two parameters α and β determine the local topology of a critical point. In the new α-β space, all the basic patterns of simple critical points are distributed on a unit circle. Feature distribution is therefore defined as a set of critical points together with their α and β parameters.

In Clifford algebra, vector fields are represented as a multiplication of linear fields, each of which is characterized by their α and β parameters. Therefore, a complex Clifford vector field is equivalent to a set of linear vector fields. This representation greatly simplifies the analysis.

The concept "work" describes the energy difference between vector fields or the amount of work required when transforming vector fields. The Earth Mover's Distance (EMD) is a linear optimization problem, it computes the minimal amount of work needed to transform one distribution into the other by moving "distribution mass". EMD here is used to compute the least amount of work necessary to convert one vector field into another. It is a quantitatively measure of similarities and dissimilarities between vector fields.

Multidimensional Scaling (MDS) computes a configuration of points in a low-dimension Euclidean distance to match the distance in the higher dimension as close by as possible. A vector field with a set of pairs of α and β values corresponds to a set of points in α-β space. MDS in this dissertation is used to sort through a list of vector fields and produce a two-dimensional map that shows the relationship (distance) between pairs of vector fields.

Feature comparisons using EMD is very useful when navigating through a large data base looking for vector fields with similar topology or comparing results from experiments and computer simulations. It is ideal for data compression of a large flow field, since only the number and types of critical points along with their corresponding α and β parameters are necessary to reconstruct the whole field. It can also be used to better quantify the changes in time varying data sets.

5.1.2 Decomposition of a Tensor Field

A second-order tensor usually has nine independent variables, even a symmetrical one has six. Therefore, a set of data from a three-dimensional second-order tensor field contains an enormous amount of information. Simplification of data before analysis becomes necessary.

In this dissertation, a decomposition of tensors into a deviator and an isotropic tensor is introduced and serves as a preprocessor. The eigen system of a deviator is parallel to that of its associated tensor. It carries the essential information of a tensor field. An isotropic tensor serves as a uniform bias. Tensor invariants of a deviator describe the existence conditions of hyperstreamlines. Degenerate points are basic constituents of a tensor field. A degenerate point is also a singular point in a deviator. A control function is obtained from the characteristic equation of a deviator and it determines the occurrences of singular points in a deviator. The representations of singular points, both double and triple, can be derived from the control function.

These singularities can further be linked to important physical properties of the underlying physical phenomena. For example, we show that for a deformation tensor in a stationary flow, the singularities of its deviator actually represent the area of the vortex core in the field; for a stress tensor in solids, the singularities represent the area with no stress; for a Newtonian flow, compressible flow and incompressible flow as well as stress and deformation tensors share similar topological features due to the similarity of their deviators; for a viscous flow, removing the large, isotropic pressure contribution enhances dramatically the anisotropy due to viscosity.

5.1.3 Topological Structure of 3D Tensor Fields

Topological representations of tensor fields reveal the geometric pictures that represent tensor data locally as well as globally in a very simple manner. This dissertation

studies the topological structure of 3D tensor fields, an issue which hasn't been dealt with before. The classification of degenerate points in 3D tensor fields is extended from that of 2D. Locally, separating surfaces divide the field around 3D degenerate points into several building blocks which are the fundamental elements in tensor fields. The separating surfaces have a general structure as they could appear at various angles as compared to 2D tensors which are confined on a plane. Each of the surfaces are characterized by patterns similar to those of hyperbolic or parabolic sectors and is bounded by hyperstreamlines that are emanating from or terminated at the degenerate point. Consequently, a point of triple degeneracy can be classified by the number and type of separating surfaces surrounding it. The trajectories on the surfaces are locally 2-D, while off the surfaces they are fully 3-D. The set of degenerate points and their connecting separatrices form the topological skeletons that depict the geometric structure of the field without visual clutter.

5.1.4 Representations of Degenerate Points

Previous research efforts on tensor field topology only focused on isolated degenerate points. However, in 3D tensor fields, points of double degeneracy often appear as continuous lines and surfaces. This dissertation designs a method that derives the fully 3D representation from a control function of a deviator which works for general cases including both triple and double degeneracy.

5.2 Analysis Framework

Figure 5.1 and Figure 5.2 are two flow charts that summarize analysis processes of the vector field comparisons and 3D tensor field topology, respectively.

Given n vector fields in a data base for comparison, Figure 5.1 starts with two input vector fields v1 and v2 and the clifford analysis procedure computes the feature

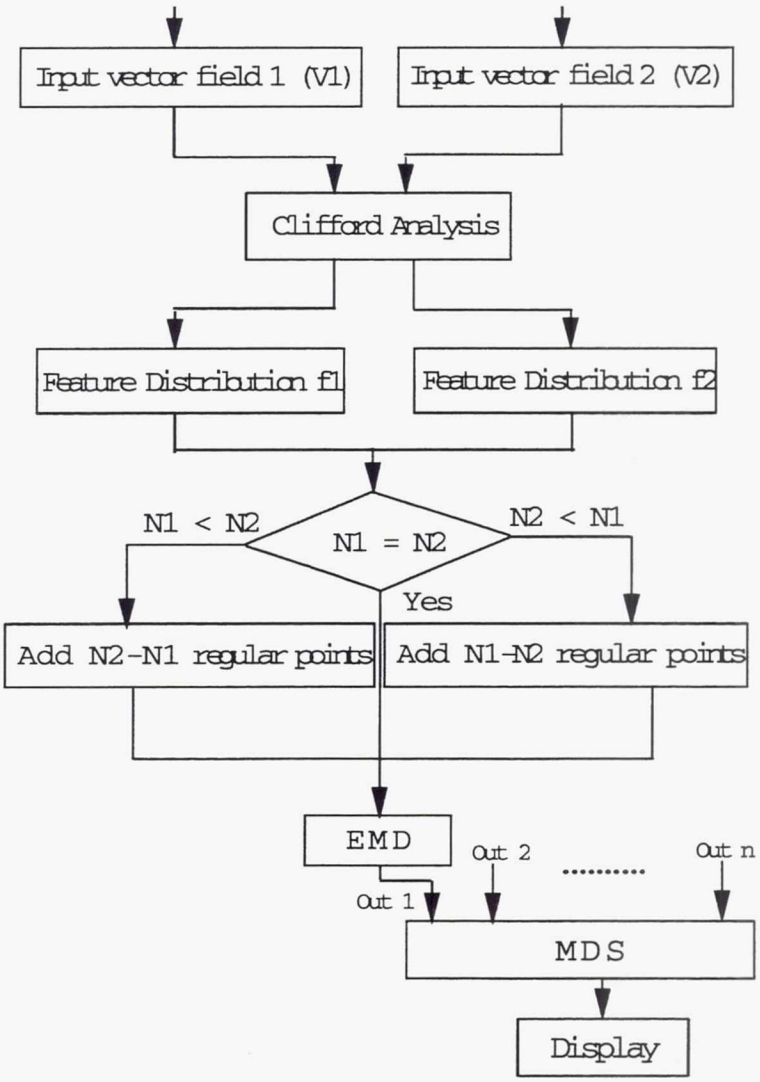


Figure 5.1: A frame work for feature comparisons of vector fields.

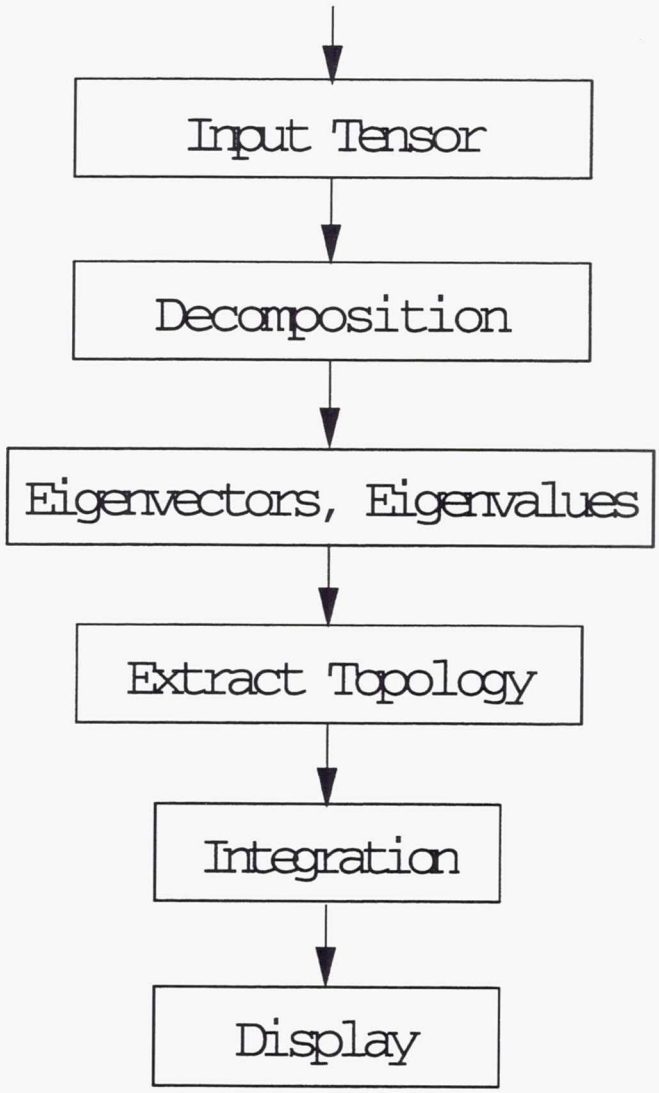


Figure 5.2: A frame work for tensor field analysis.

distributions of these two vector fields, respectively. After obtaining feature distributions f1 and f2, the next step is to compare whether these two distributions have the same number of critical points. If $N1 > N2$, $N1 - N2$ regular points **1** with $\alpha = 0$ and $\beta = 0$ are added to f2; if $N1 < N2$, $N2 - N1$ regular point **1** with $\alpha = 0$ and $\beta = 0$ are added to f1. EMD is computed when the number of critical points in f1 and f2 are even. Similarly, the EMDs between all the pairs of n vector fields are computed. Then, MDS is performed and the n vector fields are displayed on a 2D map.

Figure 5.2 describes the procedure for tensor field analysis. First, a tensor field is given as an input for the analysis. The decomposition step is a preprosessor to remove the uniform bias and obtain the deviator that describes the essential information of the input tensor. Eigen system is computed and degenerate points are located in step 3. Then, the degenerate points are analyzed and their local topological structure are extracted. The integration step generates hyperstreamlines that connect the set of degenerate points and thus forms a global topological skeleton. In the end, a simple and compact representation of this input tensor is displayed.

5.3 Future Research

Vector and tensor fields are very rich in information. Current research covers only a fraction of them. There are still a lot of interesting issues for further exploration.

- **Feature comparison for 3D vector fields**

  Currently, the method for feature comparisons only deals with 2D vector fields. A 3D critical point behaves similarly as a 2D critical point on the planes spanned by the pairs of eigenvectors of its Jacobian matrix. Therefore, it is possible to apply the feature comparison method to the 3D vector fields in a similar manner.

- **Extension of feature comparisons for tensor fields**

  A tensor field is equivalent to its eigen system. Eigenvector fields are essentially bi-directional vector fields, therefore, it is feasible to extend feature comparison methods to tensor fields.

- **Addition of other characteristics of vector fields as criteria for feature comparisons**

  Feature distributions used for comparison are not restricted to only topological characteristics. Other properties of vector fields can be used as criteria as well depending on the applications. For example, the locations of critical points in a flow are sometimes also important and are required to be part of the feature distribution. How to put totally different features together as one feature distribution set and assign different weights to them in order to rank the importance of these features is a particularly chanllenging problem.

- **Numerical solutions for locating double degeneracy**

  Representations of double degeneracy derived from control functions are still analytical solutions. Efficient computational methods to locate continuous lines and surfaces of double degeneracy will become necessary when dealing with real physical problems, both simulated and experimental.

- **Further classification for triple degenerate points**

  Patterns of 3D degenerate points presented here probably do not exhaust all the possibilities, and further classification maybe necessary for the prediction of 3D topological structures.

- **Application of tensor field analysis to real physical problems** Degenerate points have very important physical meanings. For example, in the Boussinesq problem, the triple degenerate points are stress free points and in rotational

flow, the vortex core area is also an area of triple degenerate points of the deformation tensor. It will be very interesting and useful to apply these important properties to the actual physical systems for sovling problems such as vortex core detection.

Appendix A

Related Publications

The work presented in this dissertation has appeared or will appear in several publications; this appendix contains a brief overview of the relevant articles.

- Yingmei Lavin, Rajesh Batra and Lambertus Hesselink, "Feature comparisons of vector fields using Earth Mover's Distance", *Will appear in Proc. IEEE Visualization '98*

- Yingmei Lavin, Rajesh Batra and Lambertus Hesselink, "Vector Field Comparisons using Earth Mover's Distance", *Will appear in Proc. SIGGRAPH'98, Sketches*

- Yingmei Lavin, Yuval Levy and Lambertus Hesselink, "Singularities in Nonuniform Tensor Fields," in *Proc. IEEE Visualization '97,* pp. 59-66, CS Press, Los Alamitos, CA., October 1997.

- Lambertus Hesselink, Yuval Levy and Yingmei Lavin, "The Topology of Symmetric, Second-Order 3-D Tensor Fields," *IEEE Computer Graphics and Applications,* March 1997. Special issue on scientific visualization.

- Yingmei Lavin, Yuval Levy and Lambertus Hesselink, "The Topology of Symmetric Tensor Fields," in *Proc. IEEE Visualization '96, Late Breaking Hot Topics,* pp. 43-46, CS Press, Los Alamitos, CA., 1996.

- Yingmei Lavin, Yuval Levy and Lambertus Hesselink, "The Topology of Three-Dimensional Symmetric Tensor Fields," in *13th AIAA Computational Fluid Dynamics Conference,* 1997.

Bibliography

[1] R. Hamming, *Numerical methods for Scientists and Engineers.* McGraw-Hill, New York, 1962.

[2] B. McCormick, T. DeFanti, and M. Brown, "Visualization in scientific computing," *Computer Graphics*, vol. 21, Nov. 1987.

[3] G. Nielson, H. Hagen, and H. Müller, eds., *Scientific Visualization*, ch. 1. IEEE Computer Society, 1997.

[4] R. Bergeron, W. Cody, W. Hibbard, D. Kao, K. Miceli, L. Treinish, and S. Walther, "Database issues for data visualization: Developing a data model," in *IEEE Visualization'93 Workshop on Database Issues for Data Visualization* (J. Lee and G. Grinstein, eds.), Springer-Verlag, 1995.

[5] R. A. Earnshaw and D. Watson, eds., *Data Visualization-Has It All Been Seen Before? Animation and Scientific Visualization-Tools and Applications*, ch. 1. Academic Press, 1993.

[6] P. R. keller and M. M. Keller, *Visual Cues, Practical Data Visualization.* IEEE Computer Society Press, Los Almos, Calif., 1993.

[7] P. Buning and J. Steger, "Graphical and flow visualization in computational fluid dynamics," in *AIAA 7th Computational Fluid Dynamics Conference*, June 1985. AIAA Paper 85-1507-CP.

[8] J. Helman and L. Hesselink, "Automated analysis of fluid flow topology," in *In Three-Dimensional Visualization and Display Technique, SPIE Proc.*, vol. 1083, Jan. 1989.

[9] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *Computer*, vol. 22, pp. 27–36, Aug. 1989. Also appears in *Visualization in Scientific Computing*, G. M. Nielson & B. Shriver, eds. Companion videotape available from IEEE Computer Society Press.

[10] J. L. Helman and L. Hesselink, "Visualization of vector field topology in fluid flows," *IEEE Computer Graphics and Applications*, vol. 11, no. 3, pp. 36–46, 1991.

[11] T. Delmarcelle and L. Hesselink, "Visualization of second-order tensor fields and matrix data," in *Proc. IEEE Visualization '92*, pp. 316–323, CS Press, Los Alamitos, CA., 1992.

[12] T. Delmarcelle and L. Hesselink, "Visualizing second-order tensor fields with hyperstreamlines," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25–33, 1993.

[13] T. Delmarcelle and L. Hesselink, "A unified framework for flow visualization," in *Computer Visualization* (R. Gallagher, ed.), ch. 5, CRC Press, 1994.

[14] Y. Lavin, Y. Levy, and L. Hesselink, "The topology of three-dimensional symmetric tensor fields," in *Late Breaking Hot Topics IEEE Visualization '96*, pp. 43–46, CS Press, Los Alamitos, CA., 1996.

[15] Y. Lavin, Y. Levy, and L. Hesselink, "Singularities in nonuniform tensor fields," in *IEEE Visualization '97 Proc.*, pp. 59–66, CS Press, Los Alamitos, CA., 1997.

[16] A. I. Borisenko and I. E. Tarapov, *Vector and Tensor Analysis with Applications.* Dover Publications, New York, 1979.

[17] M. Henle, *A Combinatorial Introduction To Topology.* Dover Publications, Inc., New York, 1994.

[18] L. Mach, "Über die sichtbarmachung von lufstromlinien," *Z. Luftschiffahrt Phys. Atm.*, vol. 6, pp. 129–139, 1896.

[19] W. Merzkirch, *Flow Visualization.* Academic Press, New York, 1974.

[20] L. Hesselink, "Digital image procssing in flow visualization," *Annual Review of Fluid Mechanics*, vol. 20, pp. 421–485, 1988.

[21] L. Hesselink, J. Helman, and P. Ning, "Quantitative image processing in fluid mechanics," in *Proc. of the 2nd World Conference on Experimental Heat Transfer,Fluid Mechanics and Thermodynamics*, 1991.

[22] A. P. M. Chong and B. Cantwell, "A general classification of three-dimensional flow fields," *Physics of Fluids A*, pp. 765–777, 1990.

[23] J. L. Helman and L. Hesselink, "Analysis and visualization of flow topology in numerical data sets," *Proc. of the IUTAM Symposium on Topological Fluid Mechanics*, pp. 361–371, Aug. 1990.

[24] J. L. Helman and L. Hesselink, "Surface representation of two- and three-dimensional fluid flow topology," in *SPIE Proceedings*, vol. 1459, Feb. 1991.

[25] J. L. Helman and L. Hesselink, "Analysis and representation of complex structures in separated flows," in *Proc. IEEE Visualization '90*, pp. 6–13, CS Press, Los Alamitos, CA., 1990.

[26] J. Jimenez, "Kinematic alignment effects in turbulent flows," *Physics of Fluids*, pp. 652–654, 1992.

[27] R. K. Wm.T. Ashurst, A.R. Kerstein and C. Gibson, "Alignment of vorticity and scalar gradient with strain rate in simulated navier-stokes turbulence," *Physics of Fluids A*, pp. 2343–2353, 1987.

[28] S. Pope, "A more general effective-viscosity hypothesis," *Journal of fluid mechanics*, vol. 72, pp. 331–340, 1975.

[29] C. Speziale, S. Sarkar, and T. Gatski, "Modelling the pressure-strain correlation of turbulence: an invariant dynamical systems appraoch," *J. of Fluid Mech.*, vol. 227, pp. 245–172, 1991.

[30] S. Fu, M. Leschziner, and B. Launder, "Modelling strongly swirling recirculating jet flow with reynolds-stress transport closure," in *Proc. 6th Symposium on Turbulent Shear Flow, Toulouse*, 1987.

[31] B. Launder and D. Tselepidakis, *Turbulent Sheer Flows 8.* Springer-Verlag, 1993.

[32] T. Craft and B. Launder, "A reynolds stress closure designed for complex geometries," *Int. J. Heat Fluid*, vol. 17, pp. 245–254, 1996.

[33] S. Jakirlic and K. Hanjalic, "A second-moment closure for non-equilibrium and separating high- and low-re-number flows," in *Proc. 10th Symposium on Turbulent Shear Flow, Pennsylvania State University*, 1995.

[34] F. Lien, W. Chen, and M. Leschziner, "Low-reynolds-number eddy-viscosity modelling based on non-linear stress-strain/vorticity relations," in *Proc. 3rd Symposium on Engineering Turbulence Modelling and Measurements, Crete*, 1996.

[35] T. Craft, B. Launder, and K. Suga, "Prediction of turbulent transitional phenomena with a nonlinear eddy-viscosity model," *Int. J. Heat FluidFlow*, vol. 18, p. 15, 1997.

[36] K. Suga, *Development and application of a non-linear eddy-viscosity model sensitised to stress and strain invariants.* PhD thesis, UMIST, UK, 1995.

[37] D. Apsley and M. Leschziner, "A new low-re non-linear two-equation turbulence model for complex flows," in *Proc. 11th Symposium on Turbulent Shear Flows, Grenoble*, 1997.

[38] T. Delmarcelle, *The Visualization of Second-Order Tensor Fields.* PhD thesis, Stanford University, 1994.

[39] K. A. Kroos, "Computer graphics techniques for three-dimensional flow visualization," in *Frontiers in Computer Graphics (Proc. Computer Graphics Tokyo'84)* (T. L. Kunii, ed.), Springer-Verlag, 1985.

[40] W. J. Schroeder, C. R. Volpe, and W. E. Lorensen, "The stream polygon: a technique for 3-D vector field visualization," in *Proc. IEEE Visualization '91*, pp. 126–132, CS Press, Los Alamitos, CA., 1991.

[41] W. J. Schroeder, W. E. Lorensen, G. Montanaro, and C. Volpe, "Visage: an object-oriented scientific visualization system," in *Proc. IEEE Visualization '92*, pp. 219–226, CS Press, Los Alamitos, CA., 1992.

[42] J. P. M. Hultquist, "Constructing stream surfaces in steady 3-D vector fields," in *Proc. IEEE Visualization '92*, pp. 171–178, CS Press, Los Alamitos, CA., 1992.

[43] C. Upson, "The visual simulation of amorphous phenomena," *Visual Computer*, vol. 1, pp. 321–326, Feb. 1989.

[44] C. U. et al; "The application visualization system: a computational environment for scientific visualization," *IEEE Computer Graphics and Applications*, vol. 9, pp. 39–42, Apr. 1989.

[45] J. V. Wijk, "A raster graphics approach to flow visualization," in *Proceedings Eeurographics'90, North-Holland, Amsterdam*, pp. 251–259, CS Press, Los Alamitos, CA., 1990.

[46] J. V. Wijk, "Spot noise-texture synthesis for data visualization," in *Computer Graphics(SIGGRAPH'91 Proceedings)*, vol. 25, pp. 263–272, July 1991.

[47] W. de Leeuw and J. V. Wijk, "Enhanced spot noise for vector field visualization," in *IEEE Visualization'97*, pp. 223–229, CS Press, Los Alamitos, CA., 1995.

[48] R. Crawfis and N. Max, "Texture splats for 3d scalar and vector field visualization," in *Proceedings Visualization'93*, pp. 261–265, CS Press, Los Alamitos, CA., 1993.

[49] J. Kajiya and T. Kay, "Rendering fur with three dimensional textures," *Computer Graphics*, vol. 3, pp. 271–280, July 1989.

[50] K. Perlin, "Hypertexture," *Computer Graphics*, vol. 3, pp. 253–262, July 1989.

[51] B. Cabral and L. Leedom, "Imaging vector fields using line integral convolution," in *Computer Graphics(SIGGRAPH'93 Proceedings)*, vol. 27, pp. 263–272, Aug. 1991.

[52] L. Forssell, "Visualizing flow over curvilinear grid surfaces using line integral convolution," in *Proceedings Visualization'94*, pp. 240–247, CS Press, Los Alamitos, CA., 1994.

[53] M. J. Lighthill, "Introduction. Boundary layer theory," in *Laminar Boundary Layers* (L. Rosenhead, ed.), pp. 46–113, Oxford University Press, 1963.

[54] A. Blaquiére, *Nonlinear System Analysis.* Academic, New York, 1966.

[55] A. Globus, C. Levit, and T. Lasinski, "A tool for visualizing the topology of three-dimensional vector fields," in *Proc. IEEE Visualization '91*, pp. 33–40, CS Press, Los Alamitos, CA., 1991.

[56] J. L. Helman, *Representation and visualization of vector field topology.* PhD thesis, Stanford University, 1998.

[57] H. Vollmers, "The recovering of flow features from large numerical databases," in *Computer Graphics and Flow Visualization in Computational Fluid Dynamics (von Karman Lecture Series 1991-07)*, von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, Sept. 1991.

[58] R. R. Dickinson, "Interactive analysis of the topology of 4-D vector fields," *IBM Journal of Research and Development*, vol. 35, no. 1, pp. 59–66, 1991. Special issue on visual interpretation of complex data.

[59] M. Tobak and D. J. Peake, "Topology of three-dimensional separated flows," *Annual Review of Fluid Mechanics*, vol. 14, pp. 61–85, 1982.

[60] G. T. Chapman and L. A. Yates, "Topology of flow separation on three-dimensional bodies," *Applied Mechanics Reviews*, vol. 44, pp. 329–345, July 1991.

[61] W. Merzkirch, *Flow Visualization.* Academic Press, London, second ed., 1987.

[62] A. A. Andronov, E. A. Leontovich, I. I. Gordon, and A. G. Maier, *Qualitative Theory of Second-Order Dynamic Systems.* John Wiley & Sons, Chichester, 1973.

[63] L. Perko, *Differential Equations and Dynamical Systems.* Springer-Verlag, New York, 1991.

[64] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. IEEE Internations Conferences on Computer Vision*, 1998.

[65] Y. Rubner, L. J. Guibas, and C. Tomasi, "The earth mover's distance, multidimensional scaling, and color-based image retrieval," in *Proceedings of the ARPA Image Understanding Workshop*, 1997.

[66] A. Tversky, "Features of similarity," *Psychophysics Review*, vol. 84, no. 3, pp. 327–352, 1967.

[67] W. Clifford, "On the classification of geometric algebras," in *Mathematical Papers* (R. Tucker, ed.), Macmillan, London, 1876.

[68] I. Porteous, *Clifford algebras and the classical groups.* Cambridge University Press, Cambridge, 1995.

[69] J. Gilbert and M. A. M. Murray, *Clifford algebras and Dirac operators in harmonic analysis.* Cambridge University Press, Cambridge, 1990.

[70] G.Sheuermann, H.Krüger, M.Menzel, and A.Rockwood, "Visualization of higher order singularities in vector fields," in *Proc. IEEE Visualization '97*, pp. 67–74, CS Press, Los Alamitos, CA., 1997.

PrimView: A Scientific Visualizer Library Guide v2.0

Rajesh K. Batra [1]
SciVis Lab
Stanford University

January 22, 1998

[1] Additional Coding: Yuval Levy
Testing: Yingmei Lavin
Advisor: Lambertus Hesselink

Contents

1 Introduction **2**

2 Primitive Class **2**

3 Using the Primitive Library **2**

4 Example **3**

5 Primitives provided **5**

6 Creating Primitives **7**

6.1 Constructor 7

6.2 I/O and versioning 7

6.3 Use of the dirty flag and Notification 8

6.4 Drawing the primitive in Open GL 9

1 Introduction

PrimView provides a graphical user interface (GUI) and a set of development libraries to be used for scientific visualization. Reminiscent to development packages such as Open Inventor (and the corresponding visualizer ivview), PrimView provides a package for the scientific visualization community. Currently, PrimView is available only on the Silicon Graphics Workstation running IRIX6.2 and above. The libraries are written in C++ and use Open GL, and ViewKit.

This document contains information required to create new and existing primitives from a developer's stand point. The data created from these libraries can then be outputted to a file and viewed using a GUI package (see PrimView User's Guide). A knowledge of C++ is required when working with these libraries.

2 Primitive Class

The Primitive class is the parent from which all primitives are created. It stores information that is useful to its children. Table 1 lists the data accessible to its children.

ID	Unique identification for Primitive type
Color	Byte array storing R,G,B color values range (0-255)
translate	x,y,z offset from primitive's origin. Initially identity.
rotation	Angle, and vector. Initially angle = 0.
dirty flag	Change of attribute → reconstruct primitive.
Name	String specifying label of particular primitive.

Table 1: Primitive Class Storage

In addition there are several methods that a developer of new Primitive children should override. These will be discussed in detail in Section 6. First, we shall discuss how to incorporate the current primitives supplied in the primitive library into your application.

3 Using the Primitive Library

Supplied with this manual is a SGI compiled library libPrimitives.so that you will link with your applications. Also included are several example programs using each of the various primitives. It is best to use these examples in conjunction with the header files included with the primitive library as learning sources.

There are 3 basics steps to using any Primitive. These steps are shown in Figure 1.

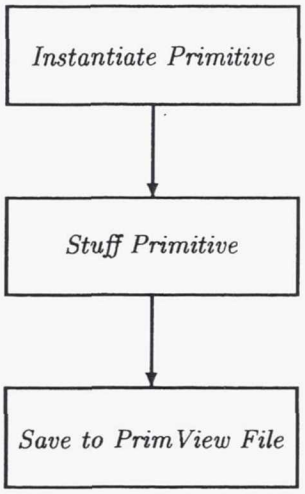


Figure 1: Steps for using the primitive class

The first step is to instantiate the primitive. Almost all primitive constructors contain optional parameters which can be filled either during creation or at a later time. A common constructor parameter is color. If no parameters are specified, default values are selected for the various constructors. In the case of colors, we attempt to select different colors for each primitive. In Section 5, default primitive colors are listed. The second step is to actually store useful information for the primitive to display. Many basic primitives just require simple information such as a position or radius. Others, such as surfaces may require several hundred triangle positions (and normals). The third step is to output the primitive to disk. All primitives support the insertion operator, `<<`. And can be outputted to a single file. Primitives will be concatenated automatically. In fact, two individual primitive files can be appended together with no problems. Every primitive is autonomous and contains information to reconstruct itself. Once the primitives are saved to disk, the data can be viewed in PrimView.

4 Example

To demonstrate the three basics steps to creating a primitive, we provide a simple example of creating some text to be displayed in PrimView. Granted, one could create the text directly in PrimView; however, using the library has the advantage of creating multiple lines of text all within the same primitive. This provides for more efficient viewing than if each line of text were to be created as a separate primitive.

The first line `Text t` instantiates the primitive. `t.add()` then proceeds to step two, and stuffs the data. Finally, `file << t` writes the data to a file. The

```
#include <fstream.h>
#include "Primitives/Text.h"

main(void)
        {
        Text t;
        t.purge(); //Wipe out any default text.
        t.add("Creating Text",0.1,.7);
        t.add("is real",0.4,.50);
        t.add("Easy!",0.6, .30);
        ofstream file("text.pv",ios::out);
        file << t;
        file.close();
        }
```

Figure 2: Example creating a Text primitive

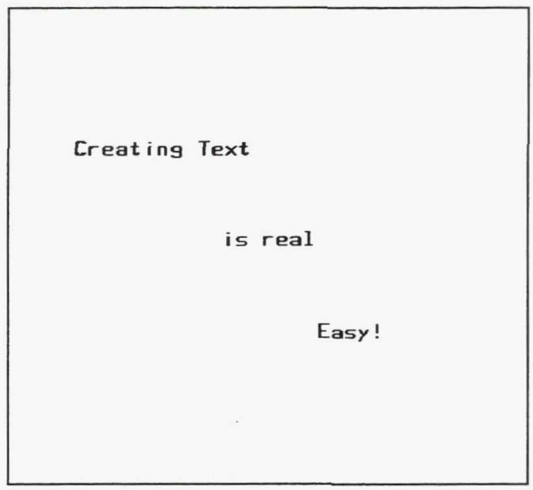


Figure 3: Output from Text primitive code (see Figure 2)

output of Figure 2, which can be viewed directly from PrimView is shown in Figure 3. Other examples are provided with the library, and follow the basic three steps.

5 Primitives provided

Currently, there are 12 basic primitives provided within this library release. Below is a description of each primitive and useful notes to keep in mind as you use them. For detailed use of the primitive see the respective header (.h) file.

Primitive: Base class which all other primitives must inherit from. Discussed in detail in Section 6.

Axes: Cartesian coordinate system with x,y,z labels. Default color is white (R=G=B=255). The coordinate system can be moved around via the `origin()` attribute, and the coordinate system can be scaled using the `size()` attribute. Both attributes can be called without parameters for querying.

Arrow: Creates an arrow with full control over its characteristics. Characteristics include head (head is where the arrow head is located) and tail location (`head`, `tail`). The size of the head radius which is the largest part of the head (it tapers to a point) can be controlled via `headRadius()`. The tail radius, which is the radius of the barrel is controlled using `tailRadius()`. The location of where the head of the arrow starts relative to the entire arrow is specified with (`headPercentage()` using a value between 0.0 to 1.0 corresponding to a percentage. The arrow's default color is green (0,255,0).

Cube: A Cube is composed of 8 points. They are specified individually using the `addElem()` method. See header file for description of order in which cube is connected. The default color for the cube is red (255,0,0). The line thickness (glLineWidth in Open GL) default is 5.0. The point size (glPointSize) is 2.0. These can be changed via `setLineSize()`, `setPointSize()`. The point color is set using the standard method `setColor()`. The line color is set using a new method `setLineColor()`.

Grid: The Grid class is a wrapper for a plot3D structured grid. Providing the grid class with a Plot3Dio class (`p3dio()`) will cause the Grid to import in a structured grid. The color by default is olive green (0,127,0), and will be drawn as a mesh with k=0 shell selected (i and j is drawn to the full span). The step size for each i,j,k slice is 1. The line thickness is default to 1.0. The grid can be rendered solid using `drawStyle()`. The line thickness used to draw the mesh is controlled via `lineSize()`. There are times when only a subsection of the grid needs to be drawn or calculated. `iMin/iMax`, `jMin/jMax`, `kMin/kMax` are attributes that control the scope of the grid.

IsoSurf: Designed to work in conjunction with algorithms such as marching cubes. Triangles are drawn in the order specified using `addElem()`. The default color for the surface is gray (95,95,95). In the constructor, a hint can be given to the IsoSurf object as to how many triangles are expected to be inputted. Internally, memory is allocated to the hint size. When memory has been exhausted by specifying more triangles than the hint, memory is reallocated to double the hint. This continues until all triangles can fit into memory. Reallocation is an expensive operation, therefore a proper hint can greatly increase efficiency.

Legend: Given a `ColorMap` is all that is needed to construct a legend. The `ColorMap` controls color space (RGB/HSV), and the minimum and maximum values to interpolate between. The Legend can be positioned using the `setOrigin()` method. The origin corresponds to the bottom left hand corner of the Legend. The coordinates are specified in 2-D and correspond to a percentage of the screen (scaled between 0.0 to 1.0). The coordinates follow a first quadrant system (i.e. origin at bottom left hand portion of the screen.)

Line: Lines are composed of a series of points. The points are added in linear fashion using the `addElem()` method. The points can be changed later using the `setOrigin()` method. (To get total number of points added to Line use the `Length()` attribute.) Lines can be drawn as a series of points and not connected (use `setConnection()`. Points can have a different color than the line itself and different point sizes (`setPointSize()`). The color of the line and its line thickness can be controlled as well (`setLineColor()`, `setLineSize()`). As a convenience function, both the line and point color can be changed simultaneously (`setColor()`). Both point-size and line-size are set to 3.0 by default. Line color is an off blue (purplish) (165,0,255).

Node: Similar to Line, however, Node can specify only one point. Nodes can be drawn as a simple Open GL point or can be rendered as a sphere (`representation()`). Default point size is set to 3.0. Default sphere radius is set to a tenth of the point size.

TexSurf: is used to render textured surfaces. The items needed for a texture are a Texture (`setTexture()`), a corresponding surface to map texture to (`setSurface()`) and a scalar value map to set each textured pixel (`setColorValues()`, `setColorMap()`). TexSurf should be used in conjunction with the TextureCalc class which is a texture calculator designed to create arbitrary surfaces and compute textures (via vector fields) over these surfaces.

Text: Designed to create text annotations. This class can add an arbitrary number of text elements at arbitrary locations via the `add()` method. Coordinate locations are specified in 2-D and are a percentage of the screen coordinates (between 0.0 to 1.0). The bottom left hand corner is

the origin. Text sentences are referenced from the bottom left hand corner as well.

Tube: The tube class is a general class creating arbitrary tubes in 3-D space. A tube is created from a series of ellipses each possessing a color. The tube merely stitches the ellipses together and linearly interpolates the color between ellipses. Ellipses are added in sequential fashion using the `addElem()` method. The resolution of the ellipses can be controlled after the ellipses have already been added (`setEllipsesResolution()`). Ellipses contain a scalar value which is interpolated and mapped to a color using `ColorMap()`. The color map can be changed at any time. In addition, the ellipses' scalar values can be overridden by setting the `colorStyle()` attribute to `UNIFORM`. The color specified via `setColor()` will be used.

6 Creating Primitives

All new primitives created by the user must inherit from the base Primitive class. This parent class provides header information (on a primitive by primitive bases) that makes your primitive recognizable throughout the system.

6.1 Constructor

When creating a new class, a unique primitive ID needs to be registered for the primitive. In the file `PrimitiveID.h` is a table of enumerations. New primitives must be *appended* to this table. It is recommended that new primitives from a particular group/agency start at a unique base i.e. `enum ePrimitiveID {... TEXT, DOD_SURFACE=200}`. If the primitive is useful and available for others, it can be incorporated into the base library and distributed as such.

The remaining parameters in the Primitive constructor are as follows:

```
Primitive(ePrimitiveID=PRIMITIVE, const char *name=NULL,
          unsigned char R=95, unsigned char G=95,unsigned char B=95);
```

The parameters all have non-essential, but highly useful data elements. A name is useful for identifying the primitive to a user. This name is what is shown in PrimView (member data `_primName`). The color parameters should be used to specify the color of the primitive (member data `_color[3]`).

6.2 I/O and versioning

When users invoke the insertion and extraction operators (>>,<<), calls to `readIn()`, `writeOut()` are made. By default, the Primitive's `readIn()` and `writeOut()` are invoked. The data written/read by this default are the primitive ID, the version number of this primitive class, the name (in binary), color, and local rotation and translation vectors. The version number is acquired by

```
void Primitive::setColor(unsigned char R,
                         unsigned char G,
                         unsigned char B)
        {
        _color[0] = R;
        _color[1] = G;
        _color[2] = B;
        _dirty = 1;
        notify();
        }
```

Figure 4: Example of using `notify` and the `_dirty` flag

invoking the method `version()`. Hence, developers are encouraged to override this attribute to return a version for their primitive. In addition, another attribute `oldestVersion()` should be overridden to support the oldest version that can be loaded by the class. Ideally, we would like backward compatibility. Using the `oldestVersion()` method is used by the default `readIn()` method to determine whether the primitive can be loaded into the system. If it can't, the read process is aborted with istream's fail bit set. If the version information is valid, the `readIn(istream &in, int version)` method is invoked. The developer should then use the version information to load the appropriate values (and set default values for information not specified in an older version).

6.3 Use of the dirty flag and Notification

The primitive class uses a simple notification system. Other classes using the primitives can register itself to be notified any time a change is made within a primitive. Therefore, in order to correctly reflect any changes made within a primitive (typically screen redraws) it is important to invoke the `notify()` method. To clarify this point let's look at the `setColor()` method shown in Figure 4.

Once the color has been stored, the `_dirty` flag is set to true, and the function `notify()` is invoked. The `notify()` method informs any registered users that a change has been made to the primitive. A typical registered user is PrimView's View engine. Once the View is notified that a change has been made to the primitive, it invokes the primitive's `drawThySelf()` method. The `_dirty` flag is internally used for efficiency. For example, in the `drawThySelf()` method, the typical Open GL construct is to use call lists. This greatly increases efficiency by invoking the call list instead of recomputing the primitive. However, when a change has been made to the primitive (color, radius, etc.) it maybe necessary to recompute the primitive. Using the `_dirty` flag informs the `drawThySelf()`

method that recomputation is necessary. Of course, once the computation is complete, the `_dirty` flag should be set to false.

6.4 Drawing the primitive in Open GL

Since the `drawThySelf()` method uses Open GL to draw itself - developers should keep the template shown in Figure 5 in mind.

```
void YourClass::drawThySelf(void)
        {
        //Enter a local coordinate system
        glPushMatrix();

        //Apply any local coordinate transformation
        glRotatef(_rotate[0],_rotate[1],_rotate[2],_rotate[3]);
        glTranslatef(_translate[0],_translate[1],_translate[2]);

        //Check if dirty flag set. If so recompute primitive.
        if (_dirty)
                {
                //Clear the dirty flag
                _dirty = 0;
                //Use Open GL commands to create a call list
                //_iDisplay contains the old display list ID.
                if (_iDisplay) glDeleteLists(_iDisplay,1);
                _iDisplay = glGenLists(1);
                glNewList(_iDisplay, GL_COMPILE_AND_EXECUTE);
                        {
                        // YOUR CODE HERE.
                        }
                glEndList();
                }
         else
                {
                glCallList(_iDisplay);
                }
        glPopMatrix();
        }
```

Figure 5: Template for `drawThySelf()` method

Index

`drawThySelf`, 8

Arrow, 5
Axes, 5

color, 3
Cube, 5

dirty flag, 8

Grid, 5
GUI, 2

I/O, 7
IsoSurf, 6

Legend, 6
library, 2
Line, 6

Node, 6
Notifier, 8

Open GL, 2
Open Inventor, 2

Primitive
    appending, 3
    constructor, 7
    ID, 7
    redraw, 9

SGI, 2

TexSurf, 6
Text, 6
Tube, 7

versioning, 7
ViewKit, 2

PrimView: A Scientific Visualizer
User's Guide
v2.0

Rajesh K. Batra [1]
SciVis Lab
Stanford University

January 22, 1998

[1] Additional Coding: Yuval Levy
Testing: Yingmei Lavin
Advisor: Lambertus Hesselink

Contents

1 Introduction **2**

2 Starting *PrimView* **2**

3 *PrimView* Window Components **2**

4 View Region Menu Options **4**

5 Option Region Menu Options **5**

5.1 Hide, Show, Remove... 5

5.2 Edit... 5

5.3 Create ▷ 7

1 Introduction

PrimView provides a graphical user interface (GUI) and a set of development libraries catered for scientific visualization. Reminiscent to development packages such as Open Inventor (and the corresponding visualizer *ivview*), *PrimView* provides a package for the scientific visualization community. Currently, *PrimView* is available only on the Silicon Graphics Workstation and runs on IRIX6.2. The libraries are written in C++ and exploit Open GL, and ViewKit.

This document contains information required to run the Graphical User Interface portion of *PrimView*. Users wishing to learn how to create data to be viewed by the GUI should refer to the library guide .

2 Starting *PrimView*

PrimView can be started with or without a command line argument. Starting without an argument loads the basic GUI interface with no data. An optional argument can be given which specifies a *PrimView* file (typically denoted with a .pv file extension).

On startup, the camera (which is what the user is looking through) points down the negative z-axis. The camera is situated a fixed percentage away from the bounding sphere of the loaded primitive. As additional primitives are added to the world (or scene) the bounding sphere is recomputed. However, the camera is not reset if the world contains at least one primitive. To reset the camera to contain all elements in the world see the **pan** function in Section 3. As primitives are added a bounded box primitive is continually updated to encompass all primitives in the world. This primitive is normally hidden but can be displayed (see Section 5 for displaying the bounding box).

3 *PrimView* Window Components

Figure 1 shows the major areas of the *PrimView* window. A brief explanation of each region follows:

View Provides a depiction of the primitives contained within the world.

- **Left mouse button** simulates a trackball and rotates the view about its own local coordinates.
- **Middle mouse button** pans or translates the world.
- **Left and Middle button** Holding both the left and the middle mouse button and moving the mouse up causes the world to zoom out. Sliding the mouse down while holding both buttons causes the world to zoom in. Note that each region is context sensitive and therefore, all mouse clicks must be within the view window for these actions to occur.

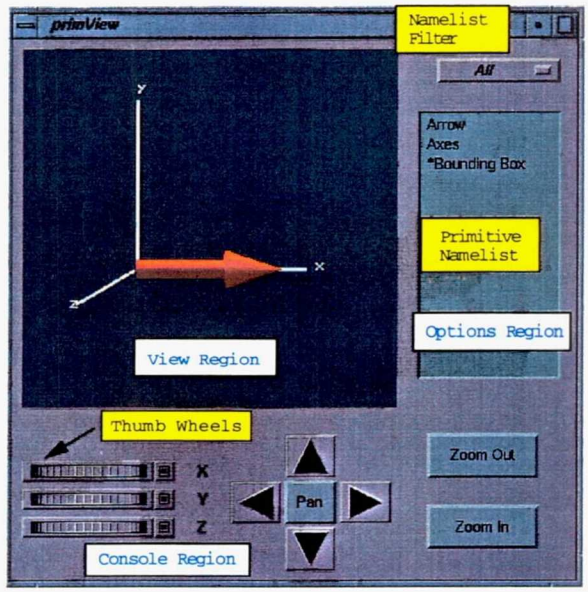


Figure 1: *PrimView* window

Right Mouse Button Provides options that manipulate the view. These menu choices are elaborated in Section 4.

Console Provides transformations on the **View** region.

X Y Z spindles Rotates world (located in **View**) about screen X,Y, and Z axes. See **View** for local rotations.

Pan buttons Translates world in screen X, and Y coordinates. Pressing the button labelled *Pan* causes the camera to reset the view.

Zoom In/Out Zooms the world by manipulating the camera.

Options Controls which primitives are shown in the **View**, and allows for editing of individual primitives.

Primitive Namelist is the region that contains the names of all of the primitives loaded in the world. Names preceded by an asterisk () indicate that the primitive is hidden and will not appear in the view.

Filter List Menu filters the list of items depicted in the **Primitive Namelist** region by primitive type. Selecting *All* shows all primitives in the **Primitive Namelist** region.

Right Mouse button Manipulates the primitives within the world. See Section 5 for details.

4 View Region Menu Options

Pressing the right mouse button over the View region yields a menu that provides manipulation of the View and the world (scene). Following is a description of each menu item found in the View menu with a brief description.

New... purges all primitives from the world, and resets the view as described in Section 2.

Open... adds primitives found in primitive file to existing world. If the primitives are older versions, *PrimView* will upgrade them to the most current version. Camera view remains unchanged as primitives are added..

Import FAST grid... This command will detect and load four types of *Structured* grids. Single-Zone, Single-Zone with I-Blank, Multi-zone (structured), and Multi-zone (structured) with I-Blank. The imported grid is converted to a Grid primitive which allows additional attributes to be changed and saved (name, color etc.)

Save Saves all *visible* primitives in the view region to disk. The filename used under the **saveas...** menu option is used. If no filename has been specified, the **saveas...** option is automatically invoked.

Save As... Saves all *visible* primitives in the view region. A file requester prompts for the filename.

Save RGB... Saves current view region as an SGI RGB picture file. The size of the RGB file is exactly that of the view region.

Save View... The camera angle, position, frustum and background color settings are saved to disk. This allows particular camera views to applied to the view region.

Load View... Loads the camera, frustum and background color settings and applies it to the view.

Background Color Brings up a color wheel, allowing the background color of the view region to be changed.

Reset View Has the same affect as pressing the Pan button in the console. Resets the camera such that the camera is placed along the z-axis and is pointing in the negative direction.

Options Toggles the visibility of the Option Region (see Figure 1).

Show Console Toggles the visibility of the Console Region (see Figure 1).

Full View Toggles between hiding both the Console and Option Region to the current Region view.

Perspective When toggled on (yellow box visible) perspective view is on (foreshortening of lines), else Orthographic projection is set.

Lighting Lighting produces shading of primitives. Without lighting pixels colors are fully saturated and uniform.

Two Sided Lighting Primitives whose normals are not pointing in the same direction as the light yield dark regions. Two Sided Lighting defeats this by lighting both sides equally. In effect as if to put a light 180 degrees to current light.

5 Option Region Menu Options

All primitives that are in the world (scene) can be controlled via the Option Menu. Individual primitives can be selected by holding down the control key and pressing the left mouse button over each primitive name located in the scroll window. Pressing the right mouse button drops a list of options for manipulating the selected items.

5.1 Hide, Show, Remove...

Hide hides primitives from the View Region. Primitives are still in memory and are denoted by an a prefixed asterisk (*) in their names shown in the Option scroll window.

Show shows all selected hidden primitives in the View Region.

Remove... On confirmation, deletes selected primitives from world (freeing up memory).

5.2 Edit...

PrimView provides some basic editing capabilities on the primitives loaded within the world (scene). Elements that have been selected can be edited by selecting the Edit... menu option. In many cases, selecting multiple primitives allows the changing of each primitive's attributes collectively. For example, to change the name of all arrows to 'Force Vectors', one can select the Arrow option in the filter menu, select all arrows with the Control key, and then select Edit from the Option Menu. Primitives *not* of the same type can not be edited together.

By in large, the majority of the primitives allow their name and color to be changed. Exceptions are noted below. In the Edit form, the Okay button accepts all changed and closes the Edit form. The Apply button merely previews the changes. To make the changes permanent the Okay button needs to be pressed. To cancel the changes made by the Apply button, press the Cancel button. This too causes the Edit form window to close.

Below is a list of the various primitives that can be edited and notes to help clarify the edit form.

Arrow : Arrows are used typically to identify locations of interest. The Edit form allows the arrow's head and tail to be manipulated similar to a vector. In addition, the Arrow Head Radius can be changed as well as the Tail's radius. The starting point of the Head can be specified as a percentage (between 0.0 (no head) to 1.0 (all head))

Axes The position of the Axis and the size can be controlled via this form. The size is useful for scaling purposes.

Cube Not implemented in this version.

Grid Allows dynamic control over which grid slice is currently displayed. This is controlled via the min/max sliders and the corresponding selected grid slice toggle. For example, to control the 'k' slice of a grid, make sure the 'k' toggle is selected and slide the min or max slider. On low performance machines, the min and max can be inputted directly in the form and either the Apply or Okay button can be pressed for the change to take place. The Step controls the coarseness of the grid. For example, selecting a step size of 2 would draw every other point to form the grid. This is useful for low performance machines. A grid can be displayed either as a mesh or as a shaded surface. This is controlled by the *View As* toggles.

Isosurface Currently the name and color are the only properties that can be modified.

Legend The Legend is considered a separate primitive and is not linked to the primitive that spawned it. The position and scale size can be controlled. The X and Y value vary between 0 to 1 and correspond to screen position of the legend's bottom left corner. (0,0) corresponds to the bottom left hand corner of the screen and (1,1) is the upper right hand corner.

Line Currently not implemented.

Node Nodes can be represented as points or as spheres. In either case the position and scale size can be modified.

Texture Color cannot be modified. Depending on the capabilities of the display machine hardware or software texture mapping can be selected. For systems supporting hardware texture mapping set the toggle to Texture Mapping. For systems that do not have hardware texture mapping, textures are rendered using triangles when the Triangle Rendering toggle is selected. Note: even on platforms which do not support hardware texture mapping, OpenGL will internally render the texture in software if the Texture Mapping toggle is set. Tests have indicated that Triangle Rendering has better performance than OpenGL's software texture mapping.

Text The position of the text's bottom left hand corner can be controlled in screen coordinates (0,0 - bottom left hand corner of screen 1,1 - top right hand corner of screen) via the Origin. The Content field changes the text displayed in the View region. The Name is merely a tag used to identify the text in the Option's Scroll window.

Tube Tubes can either be single colored or by representing a scalar value at each ellipse can depict multiple colors. Colors are formulated by interpolating within either an RGB or HSV colorspace. By default the minimum value is assigned blue and the maximum value assigned magenta. Colors in both spaces span the following spectrum: Blue, Cyan, Green, Yellow, Red, and Magenta. Under the Color Control field a tube's maximum and minimum value can be set. By changing this range, the corresponding color space interpolation changes. Colors used in the spectrum can be modified by selecting the spectrum option located in the color control region. Note that the spectrum must maintain its order i.e. a tube can span Green (min) to Yellow (max) but never Yellow (min) to Green (max). Selecting HSV or RGB selects which color space to interpolate in. RGB typically yields a much more discrete space best for scientific visualization. HSV tends to smooth the colors into one another and typically yields a more aesthetic representation. Selecting Single Color... overrides the tubes scalar data and displays the tube as a single color. The scalar data is not lost, and can be recovered by pressing the Unify Tubes... button. Unify Tubes... is typically used when multiple tubes have been selected. Its function is to set all of the selected tubes to a single minimum and maximum corresponding to the selected tubes' overall minimum and maximum values. This is useful when a uniform color map is needed over a set of tubes. Spawn Legend will create a legend primitive based on the minimum and maximum value as well as the color space found within the form. A tube's corresponds to the number of points used to draw each ellipse. The scale affects the overall radius of the tube. Note: Anytime the scale is changed, the value reverts to 1. This indicates that the current scale is now 1 and any additional change will affect the current tube's scale.

5.3 Create ▷

Certain basic primitives can be created directly within *PrimView*. These primitives are typically used to provide supplementary information to View Region. Currently, there are four primitives that can be created from the Option Menu: Arrow, Axes, Node, Text. Creation of any of these primitives automatically nvokes the edit primitive form.

Index

arguments, 2
Arrow, 6
asterisk, , hide, 3
Axes, 6

Background Color, 4
Bounding Box, 2

Camera, 2
color spectrum, 7
Console, 4
Create Primitives, 7
Cube, 6

Edit Primitives, 5

filter, 3
Full View, 4

Grid, 6
GUI, 2

Head Radius, 6
Hide, 5

Import,FAST grid, 4
Isosurface, 6

Legend, 6, 7
libraries, 2
Lighting, 5
Line, 6
Load View, 4

New, 4
Node, 6

Open, 4
Open GL, 2
Open Inventor, 2
OpenGL, 6
Option Menu, 4

Pan, 2
Perspective, 5
Primitive
    Pan, 2
    Rotation, 2
    Translation, 2
    Zoom, 2
Primitive Color, 5

Reset View, 4
Resolution, 7
RGB,Save RGB, 4
Rotation, 2

Save, 4
Save As, 4
Save View, 4
SGI, 2

Texture Mapping, 6
Translation, 2

ViewKit, 2

Zoom, 2

Feature Comparisons Of 3-D Vector Fields Using Earth Mover's Distance

Rajesh Batra
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305

Lambertus Hesselink
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-4035

1 Abstract

A method for comparing three-dimensional vector fields constructed from simple critical points is described. This method is a natural extension of the previous work [1] which defined a distance metric for comparing two-dimensional fields.

The extension to three-dimensions follows the path of our previous work, rethinking the representation of a critical point signature and the distance measure between the points. Since the method relies on topologically based information, problems such as grid matching and vector alignment which often complicate other comparison techniques are avoided. In addition, since only feature information is used to represent, and therefore stored for each field, a significant amount of compression occurs.

2 Introduction

Vector fields[1] are used to study phenomena in almost all areas of the physical sciences including such diverse subjects as climate modeling, dynamical systems, electromagnetism, and fluid mechanics. Hence, the analysis of vector fields has become a significant concern to the sciences, and a variety of techniques for visualizing and analyzing vector fields have been developed. However, an effective technique to quantitatively compare vector fields has not been developed. This paper addresses the issue.

A review of existing comparison techniques is first discussed. Even the most promising of these techniques lack the quantitative capabilities for automated comparisons. The properties of the two dimensional classification technique used in [1] are briefly discussed and are used to extend the classification to three-dimensional critical points. A complete categorization of 3-D simple critical points is presented and is used to redefine the EMD metric allowing for a quantitative comparison between 3-D flow fields. The paper concludes with an example demonstrating the effectiveness of the technique on a thermal convection model described by the Lorenz equations.

[1] The definition of vector fields is restricted to continuous fields or flows which are discussed in section 4.

3 Existing Comparison Techniques

A variety of comparison techniques exist for vector fields. These techniques basically fall into three general categories: Image, data, and feature extraction based comparisons. In most of these cases, comparisons are made visually [2].

Image based comparisons work on the computer generated image. Often times, a numerical data set is converted into an image that simulates an experimental visualization technique (computational flow imaging). This may be easier than extracting a vector field from an image, such as Schlieren. However, visualizing a field in 3-D is quite difficult and often, these techniques are limited to two dimensions. In addition to side-by-side comparison of images, other techniques include image fusion, and Fourier analysis [3].

Data level comparison techniques operate directly on the raw data. An accurate comparison requires proper grid alignment which can involve problematic interpolation between two fields [4].

The last comparison category is the extraction of features. Typically features are problem specific; for example in fluid mechanics features include vortex cores and shock surfaces. Often there is a geometric representation of the feature and possibly a semantic representation of the system which can be compared using a pattern recognition technique [5]. This may lead to more robust comparisons.

Qualitative comparisons have been based on the concept of critical points in vector fields. Past study has focused on the geometric structure of vector fields [6] and last year a quantitative measure for two dimensional vector fields was introduced [1]. The work is extended by defining a quantitative measure of the similarities and differences of three-dimensional vector fields.

4 Description of Phase Portraits with 2-D α-β Parameters

Since the method reduces a 3-D flow pattern into 2-D components, the relevant 2-D categorization for this type of vector field is reviewed. A 2-D vector field that can be represented as a system of two simultaneous differential equations has the following form:

$$v_x = \frac{dx}{dt} = F(x,y) \qquad (1)$$
$$v_y = \frac{dy}{dt} = G(x,y)$$

where F and G are continuous and have continuous partial derivatives in some region D. The solutions to this system forms a family of directed paths. Given some initial value to the system, a parametric representation expressed as $x = \phi(t)$, $y = \psi(t)$ can be deduced. The image formed is the phase portrait and is typically described by the number, type, and arrangement of critical points (or equilibrium points). These are points where $F(x,y) = 0$ and $G(x,y) = 0$. The nature of a critical point will not change under continuous (affine) transformation. Critical points are significant in that they are the only points in a vector field where tangent curves may cross each other. Therefore, critical points delineate the field into sectors of uniform flow.

A critical point is said to be isolated or simple if there is an open neighborhood around it that contains no other critical points. The behavior of the flow about a critical point can be analyzed by investigating the trajectories in the neighborhood of the critical point. If the distance is sufficiently small (say dx, dy), a first order approximation (Equation 2) of the field can be used.

$$v_x(dx,dy) \approx \frac{\partial v_x}{\partial x}dx + \frac{\partial v_x}{\partial y}dy \qquad (2)$$
$$v_y(dx,dy) \approx \frac{\partial v_y}{\partial x}dx + \frac{\partial v_y}{\partial y}dy$$

Hence, the flow pattern is completely determined by the Jacobian, $J_{ij} = \frac{\partial v_i}{\partial v_j}$ $(i,j = 1,2)$ evaluated at the critical point. The various patterns formed in the phase-plane space can be seen by analyzing the eigenvalues of the Jacobian. The characteristic equation

$$\lambda^2 + P\lambda + Q = 0 \qquad (3)$$

where $P = -trace(J)$ and $Q = det(J)$ is used to classify the various patterns using the well known $P-Q$ stability diagram [7]. However, advantageous properties arise by defining a new space (α', β') as explained in [1], where the eigenvalues map $\alpha = P$ and $\beta = sign(P^2 - 4Q)\sqrt{|P^2 - 4Q|}$

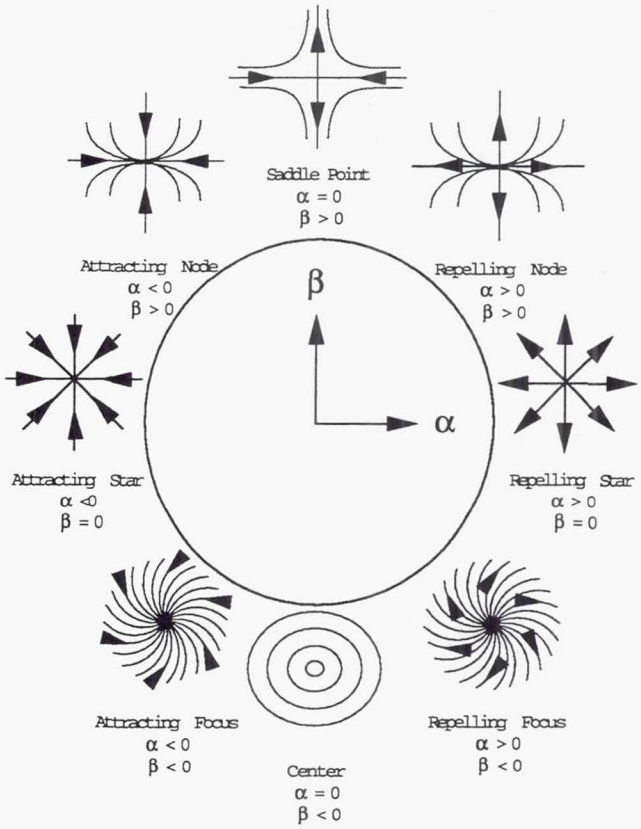


Figure 1: Basic patterns for simple critical points.

and are normalized as follows:

$$\alpha' = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}} \qquad (4)$$
$$\beta' = \frac{\beta}{\sqrt{\alpha^2 + \beta^2}} \qquad (5)$$

In this space, critical points obey all the the rules defined for a regular 2-D Euclidean space and the distance between any two critical points is a metric.

It is shown in [8] that the actual values of α and β do not determine the portrait of the critical point; only the ratio between them matters. Hence, this normalization maps all points onto a unit circle and thereby provides a means of relatively quantifying the difference between various points by just an angle. Also note that a uniform vector field with no critical points, $F(x,y) = c_1$, $G(x,y) = c_2$ has $\alpha = 0$ and $\beta = 0$ and maps to the origin of the unit circle. This is the reason why arc length is not used as a metric. For the remainder of the paper, α and β values will be assumed normalized. The patterns are sketched in Figure 1 and enumerated in Table 1 [2]. Notice a positive or negative real part

[2] The definition of saddle indicated in the table is more relaxed than spec-

α	β	Type	Constraint
>0	$=0$	Repelling Star	
>0	>0	Repelling Node	$\lvert\beta\rvert < \lvert\alpha\rvert$
—	>0	Saddle	$\lvert\beta\rvert > \lvert\alpha\rvert$
<0	>0	Attracting Node	$\lvert\beta\rvert < \lvert\alpha\rvert$
<0	$=0$	Attracting Star	
<0	<0	Attracting Focus	
$=0$	<0	Center	
>0	<0	Repelling Focus	

Table 1: Classification of Critical Points via α-β values

(denoted by α) is indicative of repelling/attracting behavior. And if an eigenvalue has an imaginary part ($\beta < 0$), it indicates circulation about the point and the trajectories can be represented via logarithmic spirals, otherwise asymptotic behavior whose trajectories can be described via simple power laws is exhibited.

5 Classification of Three-Dimensional Vector Fields Using Phase Portraits

The formulation for a 3-D vector field is very similar to the 2-D analysis. For a three-dimensional vector field, the Jacobian is represented by a $3X3$ matrix, $J_{ij} = \frac{\partial v_i}{\partial v_j}$ $(i, j = 1, 2, 3)$ The characteristic equation now becomes

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \tag{6}$$

where $P = -trace(J)$, $Q = \frac{1}{2}(P^2 - trace(J^2))$, and $R = -det(J)$. Three distinct eigenvalues are possible, along with three eigenvectors. The flow field can be decomposed into fundamental solution trajectories along its eigenvector planes as demonstrated by Reyn [9] and Chong et al. [10]. All other solutions trajectories converge (or diverge) to these eigenvector planes. Therefore a critical point in 3-D can be defined by a set of three (α, β) values. Each (α, β) point corresponds to a solution trajectory formed in the respective eigenvector plane.

To simplify the classification of the various phase portraits about a three-dimensional critical point, the Jacobian is transformed into canonical form. This does not affect the eigenvalues since they are invariant to changes in scale, translation, and rotation. Philippou, and Strickland categorized the Jacobian into seven basic canonical forms [11] or classes. With each form, several phase-portraits are possible. In Tables 2 and 3, all possible cases are enumerated in a similar style as presented in reference [11] [3] along with the

ified in the figure.

[3]The class structure is slightly changed by placing the complex Jordan form last.

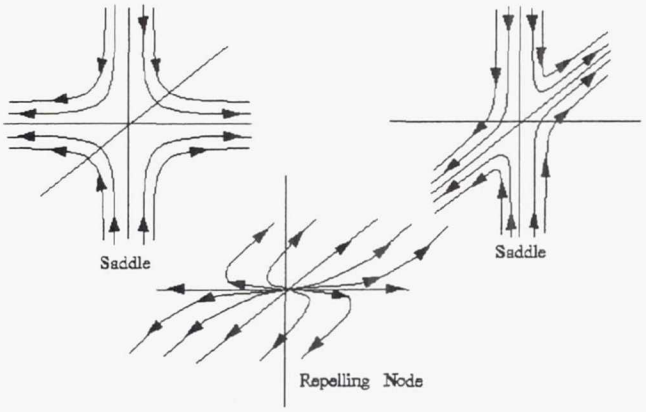


Figure 2: Decomposition of a class 1 critical point along eigenvector planes.

corresponding α, β sets. A brief discussion of the various classes is given below.

Class 1 $(\lambda_1, \lambda_2, \lambda_3)$: Class 1 is indicative of eigenvalues which are real and distinct. In this case, there exists three independent eigenvectors and therefore three independent eigenvector planes. For the case of a Hermitian matrix (and its subclass the real symmetric matrix), the eigenvectors are mutually orthogonal. If all of the eigenvalues are positive, repelling nodes form in all three planes. If the signs differ, a saddle occurs in two of the planes, and finally if all signs are negative attracting nodes occur. All other solution trajectories approach or diverge from the critical point as $t \to \infty$ and are not planar. In this case, there exists a full set of α, β values. A degenerate 2-D case is exhibited when one of the eigenvalues is zero. Only one plane will contain a simple solution trajectory, the other two planes will contain lines (since in 2-D every plane in the third dimension is identical). In this case, there exists only one distinctive α, β value. The remaining two are set to zero. In fact, all 2-D cases degenerate to $\{\{\alpha_1, \beta_1\}, \{0, 0\}, \{0, 0\}\}$ and the computed comparison values are identical to those in reference [1]. Figure 2 is an example of trajectories formed for a node/saddle/saddle combination.

Class 2 $(\lambda_1, \lambda_1, \lambda_2)$: This is a degenerate case where two eigenvalues are identical. The multiplicity is 2, however, 3 independent eigenvectors can still be found. One plane will contain a star pattern, the planes normal to this will contain solution trajectories (nodes or saddles) and will have identical α, β values. The 2-D case degenerates to the star pattern. The 1-D ($\lambda_1 = 0$) case is ignored.

Class 3 $(\lambda_1, \lambda_1, \lambda_2)$: In this case, only 2 independent eigenvectors and only two solution trajectories exist. In one plane, a log star[4] trajectory is observed and in the other plane

[4]A log star is also referred to as an improper node see [12]

a node or saddle. A log star in the α, β space is indistinguishable from a star pattern, i.e. $\alpha = \pm 1, \beta = 0$ since the star pattern is just a special case of the family of logarithmic stars formed.

Class 4 $(\lambda_1, \lambda_1, \lambda_1)$: Case 4 exhibits triple degeneracy with three independent eigenvectors. Any plane passing through the critical point will exhibit a star trajectory. The set of α, β values are identical.

Class 5 $(\lambda_1, \lambda_1, \lambda_1)$: Two linearly independent eigenvectors exist for this triple degeneracy case. Here there are only two independent planar trajectories. One trajectory is a log star located in the coordinate plane spanning the eigenvector (in x-y plane for the canonical form) and a star pattern in the other coordinate plane (x-z).

Class 6 $(\lambda_1, \lambda_1, \lambda_1)$: In this case, the multiplicity is three but only one independent eigenvector exists. Therefore, only one plane contains an attracting/repelling log star trajectory. Hence only one unique α, β value exists and this case conflicts with the 2-D case of class 2 and 3. Therefore, some false positives can be expected. Fortunately, class 1 is the most common occurrence [11], which this method classifies most uniquely.

Class 7 $(\alpha + \beta, \alpha - \beta, \lambda_3)$: Three eigenvalues are found, two of which must be complex conjugates of each other for the J matrix which contains no imaginary values. Only one plane will contain solution trajectories which are either a focus or a center. Hence, only one unique α, β pair value exists just as in the 2-D case. The real eigenvalue, λ_3, denotes a stretching or compressing phenomenon where trajectories either spiral away or towards the solution plane, and this will not be captured [13].

6 Feature Comparison via EMD

A flow field can now be described by a set of α, β values. To compare two flows, one approach is to find the closest match between the two sets of α, β values. The EMD algorithm provides this functionality. EMD is emphasized, since other techniques exist such as graph matching which takes into account connections between critical points. In the original description of EMD, terminology such as feature distribution, energy and work are used. The terminology is maintained for consistency and further information can be found in [14].

Earth Mover's Distance computes the minimal amount of work required to transform one distribution to another. In the case of vector fields, the distribution can be represented as the set of α, β values.

Definition 1 (Feature Distribution) *A feature distribution for a 3-D vector field is the set of sets of α and β values associated with the vector field's critical points:*

$$\{\{(\alpha_1^{(1)}, \beta_1^{(1)}), (\alpha_2^{(1)}, \beta_2^{(1)}), (\alpha_3^{(1)}, \beta_3^{(1)})\}, \{(\alpha_1^{(2)}, \beta_1^{(2)}), (\alpha_2^{(2)}, \beta_2^{(2)}), (\alpha_3^{(2)}, \beta_3^{(2)})\}, \ldots, \{(\alpha_1^{(n)}, \beta_1^{(n)}), (\alpha_2^{(n)}, \beta_2^{(n)}), (\alpha_3^{(n)}, \beta_3^{(n)})\}\} \quad (7)$$

Definition 2 (Energy) *The energy for a vector field is:*

$$Energy = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{3} ((\alpha_i^{(j)})^2 + (\beta_i^{(j)})^2)},$$

where n is the total number of critical points in this field.

The energy is a quantity that characterizes the critical points of a vector field. It is different from physical energy. The concept "work" is used to measure the energy differences between two vector fields or the amount of energy used to transform one vector field into the other. For a 3-D vector field, work will be defined at two levels. At the higher level, the work required to convert one set of α, β values representing a critical point into another is needed. This is denoted as the Set Work. At the lower level, the work required to convert one α, β value in the set into another is needed. This will be defined as the Elemental Work. The Set Work is therefore the minimum amount of Elemental Work required to convert one set into another. Therefore, EMD can be used on the Set Work where the distance function is the Elemental Work.

Definition 3 (Set Work) *For two vector fields with feature distributions*

$$\{\{(\alpha_1^{(1)}, \beta_1^{(1)}), (\alpha_2^{(1)}, \beta_2^{(1)}), (\alpha_3^{(1)}, \beta_3^{(1)})\}, \{(\alpha_1^{(2)}, \beta_1^{(2)}), (\alpha_2^{(2)}, \beta_2^{(2)}), (\alpha_3^{(2)}, \beta_3^{(2)})\}, \ldots, \{(\alpha_1^{(n)}, \beta_1^{(n)}), (\alpha_2^{(n)}, \beta_2^{(n)}), (\alpha_3^{(n)}, \beta_3^{(n)})\}\}$$

and

$$\{\{(\hat{\alpha}_1^{(1)}, \hat{\beta}_1^{(1)}), (\hat{\alpha}_2^{(1)}, \hat{\beta}_2^{(1)}), (\hat{\alpha}_3^{(1)}, \hat{\beta}_3^{(1)})\}, \{(\hat{\alpha}_1^{(2)}, \hat{\beta}_1^{(2)}), (\hat{\alpha}_2^{(2)}, \hat{\beta}_2^{(2)}), (\hat{\alpha}_3^{(2)}, \hat{\beta}_3^{(2)})\}, \ldots, \{(\hat{\alpha}_1^{(n)}, \hat{\beta}_1^{(n)}), (\hat{\alpha}_2^{(n)}, \hat{\beta}_2^{(n)}), (\hat{\alpha}_3^{(n)}, \hat{\beta}_3^{(n)})\}\}$$

The amount of work necessary for transforming one vector field into the other is defined as:

$$Work_{set} = \sum_{i=1}^{n} EMD_e(\{(\alpha_1^{(i)}, \beta_1^{(i)})(\alpha_2^{(i)}, \beta_2^{(i)}), (\alpha_3^{(i)}, \beta_3^{(i)})\}, \{(\hat{\alpha}_1^{(i)}, \hat{\beta}_1^{(i)}), (\hat{\alpha}_2^{(i)}, \hat{\beta}_2^{(i)}), (\hat{\alpha}_3^{(i)}, \hat{\beta}_3^{(i)})\})$$

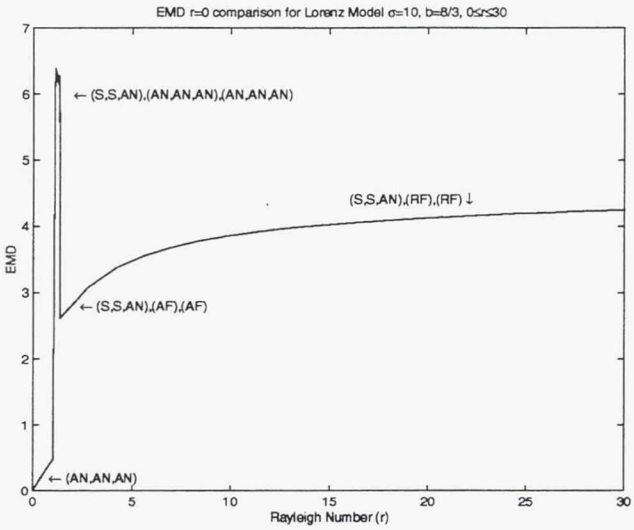


Figure 3: EMD capturing topological changes to Lorenz Model

where EMD_e is the Earth Mover's Distance whose distance function is the Elemental Work.

Definition 4 (Elemental Work) *For two vector fields with feature distributions*

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}$$

and

$$\{(\alpha'_1, \beta'_1), (\alpha'_2, \beta'_2), \ldots, (\alpha'_n, \beta'_n)\}.$$

The amount of work necessary for transforming one vector field into the other is defined as: $Work_e = \sqrt{\sum_{i=1}^{n}((\alpha_i - \alpha'_i)^2 + (\beta_i - \beta'_i)^2)}$.

Notice that the definition for elemental work is identical to the work defined for a 2-D critical point defined in [1]. Since we decompose a 3-D critical point into a set of 2-D critical points, EMD at this lower level is the same as the 2-D case. To summarize, we can find the distance between two 3-D vector fields by extracting all 3-D critical points in the fields and representing them by a set of α, β values. Earth Mover's Distance is used to find the minimum energy between the critical points just as in the 2-D case. However, the distance metric is defined as the EMD over the *set* of α, β values whose distance metric is further defined as the elemental work. The elemental work is merely the Euclidean distance in α, β space and therefore is a metric. It can be shown that the Set Work is also a metric since EMD is a metric [14].

7 Example: Lorenz Model

One application for field comparisons is in the area of meteorology. Weather patterns can be searched for in a database to understand the development of a flow. E.N. Lorenz attempted to model thermally induced fluid convection in the atmosphere using the Navier-Stokes equations [15]. Using two-dimensional motion, fluid heated from below and cooled from above under the effects of gravity produce circulation or convection rolls. This phenomenon is summarized in Equation 8.

$$\begin{aligned} x' &= \sigma(y - x) \\ y' &= rx - y - xz \\ z' &= -bz + xy \end{aligned} \tag{8}$$

x(t) represents a measure of the fluid velocity (amplitude of the convection motion), and y(t), z(t) represent measures of the spatial temperature distribution. The equations are in non-dimensional form where σ is the Prandtl number (ratio of kinematic viscosity to thermal conductivity), b is a geometric factor and r is the Rayleigh number and is proportional to the temperature difference between the upper and lower regions of the system. These equations were one of the first to demonstrate chaotic behavior, and have resulted in over a hundred papers [16]. The resulting phase portrait of the system is in three-dimensions and is composed of simple critical points.

In the original study by Lorenz σ and b were fixed, and the Rayleigh number (i.e. the temperature difference between the plates) was varied. When r is below 1 only one critical point exists. As the temperature difference increases, three critical points form and eventually for large enough r the entire system becomes unstable. Using α, β space, the behavior of the system can clearly be seen and in fact the transformation from a stable system to an unstable system is continuous in this space.

Beginning with $r = 0$, Equation 8 has one critical point at the origin. As seen in Figure 4a, the critical point is type Class I whose phase portraits are three attracting (stable) nodes. As r is increased to 1, the angle measured from the positive α axis reduces and therefore the nodes become less stable as it nears the saddle point. The arrow in Figure 4a represents the critical point's progression with increasing r. At $r = 1$, the angle is 135° and a degenerate node forms (Figure 4b). Increasing r further, the origin's phase portrait becomes a set of two saddles, and an attracting node (Figure 4c). As r increases, the saddle points approach the repelling node.

In addition, for $r > 1$, two other critical points come into existence at $(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1)$. For r near 1, another set of stable attracting nodes form (Figure 4d). However as r approaches 1.346, the angle increases

approaching an attracting star. Since the system is non-linear, the slightest perturbation causes the attracting star to become an attracting focus. Hence for $1.346 < r < 24.74$ (Figure 4e) an attracting focus forms. As r is increased to 24.76, the phase portrait changes gradually to a center. Hence, the observation of the unstable limit cycles forming around $r = 13.926$. Once r increases past 24.74, the angle increases to over 270° and a repelling focus comes into existence and the system is unstable.

The change with temperature can also be understood by observing how the phase portrait for a particular r compares with the remaining phase portraits. Figure 3 plots the EMD values required to change the critical points for $r = 0$ into other critical points at other r values. As can been seen, for $r < 1$, a gradual increase in EMD occurs as the attracting node at the origin becomes a saddle. As soon as $r > 1$, two new critical points form causing a large jump in EMD. The jump is drastic since six additional (2-D) critical points (attracting nodes) must form. Once $r > 1.346$, the phase portrait changes from three attracting nodes, to one attracting focus. Hence, less energy is required to create two additional focii than six critical points. From $r > 1.346$, the EMD value slowly increases as the system continuously moves further from a stable system to an unstable one. Not only is the attracting focii becoming a repelling focii, but the saddles at the origin are approaching degenerate repelling nodes (Figure 4c).

8 Conclusion and Future Work

We have extended the feature based comparison method to three-dimensional vector fields. We have shown that the extension can be straight-forward if we use the property that a 3-D critical point can be decomposed into a set of 2-D critical points with planar phase portraits. In addition, the redefined distance function for EMD remains a metric.

As with the 2-D case, connections between the critical points are not considered. Our experience has shown that for many cases this is sufficient, however, to reduce the number of false positives and to provide a better distance between two fields the connections should be taken into account. We are investigating this aspect along with generalizing this method to tensor fields.

This method has also demonstrated its usefulness in understanding complicated phenomenon such as the Lorenz model. The evolution of the thermal convection can be captured with EMD. Since the system is represented by quantities, fast searches can be easily constructed to locate particular patterns.

Acknowledgments

The authors are most indebted to Yingmei Lavin and David Krieger for their insightful discussions and to Yossi Rubner for providing EMD and MDS codes.

References

[1] Y. Lavin, R. Batra, and L. Hesselink, "Feature comparisons of vector fields using earth mover's distance," in *Proc. IEEE/ACM Visualization '98*, pp. 413–415, North Carolina, October 1998.

[2] F. H. Post and J. J. van Wijk, "Visual representations of vector fields: recent developments and research directions," in *Scientific Visualization* (L. Rosenblum, R. Earnshaw, and et al., eds.), pp. 181–195, Academic Press, Heidelberg, 1994.

[3] H.-G. Pagendarm and F. H. Post, "Comparative visualization-approaches and examples," in *Visualization in Scientific Computing* (M. Göbel, H. Müller, and B. Urban, eds.), pp. 95–108, Springer-Verlag, 1995.

[4] Q. Shen, A. Pang, and S. Uselton, "Data level comparison of wind tunnel and computational fluid dynamics data," in *Proc. IEEE Visualization '97*, pp. 67–74, CS Press, Los Alamitos, CA., 1997.

[5] L. Hesselink, "Digital image processing in flow visualization," *Annual Review of Fluid Mechanics*, vol. 20, pp. 421–485, 1988.

[6] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *IEEE Computer*, vol. 22, pp. 27–36, Aug 1989. Also appears in *Visualization in Scientific Computing*, G. M. Nielson & B. Shriver, eds. Companion videotape available from IEEE Computer Society Press.

[7] A. Perry and B. Fairlie, "Critical points in flow patterns," *Advances in Geophysics B*, vol. 18, pp. 299–315, 1974.

[8] Y. Lavin, *Topology based visualization for vector and tensor fields*. PhD thesis, Stanford University, 1998.

[9] J. W. Reyn, "Classification and description of the singular points of a system of three linear differential equations," *Zeitschrift für angewandte Mathematik und Physik (ZAMP)*, vol. 15, pp. 540–557, July 1964.

[10] M. S. Chong, A. E. Perry, and B. J. Cantwell, "A general classification of three-dimensional flow fields," *Physics of Fluids A*, vol. 2, no. 5, pp. 765–777, 1990.

[11] P. A. Philippou and R. N. Strickland, "Vector field analysis and synthesis using three dimensional phase portraits," *Graphical Models and Image Processing*, vol. 59, pp. 446–462, November 1997.

[12] W. E. Boyce and R. C. DiPrima, *Elemetrary Differential Equations and Boundary Value Problems*. John Wiley & Sons, fourth ed., 1986.

[13] J. M. Chacín, B. J. Cantwell, and S. Kline, "Study of turbulent boundary layer structure using the invariants of the velocity gradient tensor," *Journal of Experimental Thermal and Fluid Science*, vol. 13, pp. 308–317, 1996.

[14] Y. Rubner and C. Tomasi, "The earth mover's distance as a metric for image retrieval," Tech. Rep. STAN-CS-TN-98-86, Department of Computer Science, Stanford University, September 1998.

[15] E. N. Lorenz, "Deterministic non-periodic flow," *J. Atmos. Sci.*, no. 20, pp. 130–141, 1963.

[16] C. Stoker, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*. Springer-Verlag, 1982.

Class	Canonical Form	Notes	Phase Portraits	$\|\|\alpha\|\|$	$\|\|\beta\|\|$
1	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$	Eigenvalues are real and distinct. Eigenvectors are linearly independent.	A-Node, A-Node, A-Node R-Node, R-Node, R-Node A-Node, Saddle, Saddle R-Node, Saddle, Saddle	$\alpha_1 = \frac{\lambda_1+\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\alpha_2 = \frac{\lambda_1+\lambda_3}{\sqrt{2(\lambda_1^2+\lambda_3^2)}}$ $\alpha_3 = \frac{\lambda_2+\lambda_3}{\sqrt{2(\lambda_2^2+\lambda_3^2)}}$	$\beta_1 = \frac{\lambda_1-\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\beta_2 = \frac{\lambda_1-\lambda_3}{\sqrt{2(\lambda_1^2+\lambda_3^2)}}$ $\beta_3 = \frac{\lambda_2-\lambda_3}{\sqrt{2(\lambda_2^2+\lambda_3^2)}}$
	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	2-D case	A-Node R-Node Saddle	$\alpha_1 = \frac{\lambda_1+\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\alpha_2 = 0$ $\alpha_3 = 0$	$\beta_1 = \frac{\lambda_1-\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\beta_2 = 0$ $\beta_3 = 0$
2	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}$	Eigenvectors are independent.	A-Star, A-Node, A-Node R-Star, R-Node, R-Node R-Star, Saddle, Saddle A-Star, Saddle, Saddle	$\alpha_1 = \pm 1$ $\alpha_2 = \frac{\lambda_1+\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\alpha_3 = \frac{\lambda_1+\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$	$\beta_1 = 0$ $\beta_2 = \frac{\lambda_1-\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$ $\beta_3 = \frac{\lambda_1-\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$
	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	2-D case	A-Star R-Star	$\alpha_1 = \pm 1$ $\alpha_2 = 0$ $\alpha_3 = 0$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = 0$
	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}$	1-D case	ignore	–	–
3	$\begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}$	Eigenvalues are real. 1 linearly dependent eigenvector.	A-Log Star, A-Node R-Log Star, R-Node R-Log Star, Saddle A-Log Star, Saddle	$\alpha_1 = \pm 1$ $\alpha_2 = 0$ $\alpha_3 = \frac{\lambda_1+\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = \frac{\lambda_1-\lambda_2}{\sqrt{2(\lambda_1^2+\lambda_2^2)}}$
	$\begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	2-D case	A-Log Star R-Log Star	$\alpha_1 = \pm 1$ $\alpha_2 = 0$ $\alpha_3 = 0$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = 0$
	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}$	1-D case	ignore	–	–

Table 2: Phase Portraits for Classes 1,2,3. Legend: A- : Attracting R- : Repelling $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq 0$, $\lambda_1 > \lambda_2 > \lambda_3$

Class	Canonical Form	Notes	Phase Portraits	$\|\|\alpha\|\|$	$\|\|\beta\|\|$
4	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_1 \end{bmatrix}$	Eigenvalues are real. Eigenvectors are linearly independent.	A-Star, A-Star, A-Star R-Star R-Star, R-Star	$\alpha_1 = \pm 1$ $\alpha_2 = \pm 1$ $\alpha_3 = \pm 1$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = 0$
5	$\begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_1 \end{bmatrix}$	Eigenvalues are real. One linearly dependent eigenvector.	A-Log Star, A-Star R-Log Star, R-Star	$\alpha_1 = \pm 1$ $\alpha_2 = 0$ $\alpha_3 = \pm 1$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = 0$
6	$\begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 1 \\ 0 & 0 & \lambda_1 \end{bmatrix}$	Eigenvalues are real. Two linearly dependent eigenvectors.	A-Log Star R-Log Star	$\alpha_1 = \pm 1$ $\alpha_2 = 0$ $\alpha_3 = 0$	$\beta_1 = 0$ $\beta_2 = 0$ $\beta_3 = 0$
7	$\begin{bmatrix} \alpha & -\beta & 0 \\ \beta & \alpha & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$	Two complex and one real eigenvalues.	A-Focus R-Focus Center	$\alpha_1 = \frac{\alpha}{\sqrt{\alpha^2+\beta^2}}$ $\alpha_2 = 0$ $\alpha_3 = 0$	$\beta_1 = \frac{\beta}{\sqrt{\alpha^2+\beta^2}}$ $\beta_2 = 0$ $\beta_3 = 0$

Table 3: Phase Portraits for Classes 4-7. Legend: A- : Attracting R- : Repelling $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq 0$, $\lambda_1 > \lambda_2 > \lambda_3$

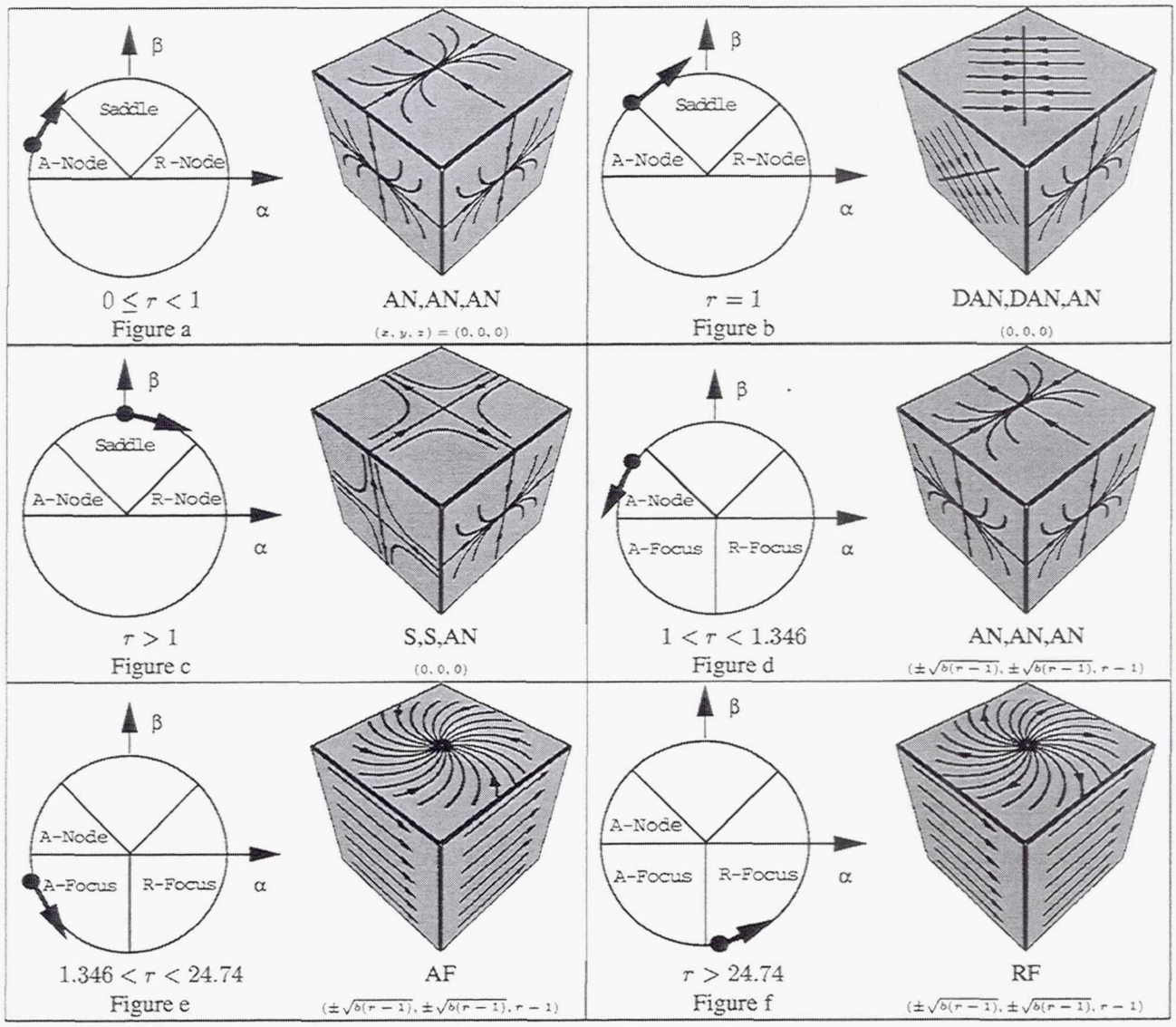


Figure 4: Lorenz Model for Fluid Convection depicted in α, β space. $\sigma = 10, b = 8/3$. AN:Attracting Node, DAN:Degenerate Attracting Node, S:Saddle, AF:Attracting Focus, RF:Repelling Focus

Topology Based Vector Field Comparisons Using Graph Methods

Rajesh Batra
Dept of Aero/Astronautics
Stanford University
Stanford, CA 94305-4035

Kerstin Kling
Inst. of Process Engineering
University of Hannover
Hannover, Germany

Lambertus Hesselink
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-4035

1 Abstract

The previous work [1] introduced a quantitative method for comparing 2-D vector fields based on the number and type of critical points that comprised the field. However, the arrangement of the critical points was ignored potentially causing two very different fields with the same type of critical points but different streamline connections to be detected as similar fields. This paper improves the comparison technique by considering the connections between critical points, thereby improving the representation of the vector field.

2 Introduction

Comparing vector fields is an important analysis technique used in the sciences. The comparison of fields aids in understanding and can improve mathematical models of phenomena. For example, a database containing historical weather patterns can be matched with the present day's pattern to better understand the evolution of the flow. In fluid mechanics, simulated flow fields can be validated with experimentally acquired data sets or with other models. Unfortunately, many of the techniques rely on rendering to a computer image and visually comparing the fields. This does not lend itself easily to automated comparison techniques, and certain fields may deceive the human eye. Data comparison techniques operate directly on the raw data but requires a common domain, grid alignment, and interpolation [2]. Feature based comparisons are often problem specific, i.e. in fluid mechanics features include shock surfaces and vortex cores. These features are typically qualitatively compared.

Recently in [1], a method for quantitatively comparing vector fields based on topological features was introduced. A vector field was characterized by the number and type of critical points. The pattern formed about the critical point was uniquely represented by an (α, β) coordinate. One of the properties of the (α, β) space was that Euclidean distances were valid and could be used as a difference measure between two critical points. Intuitively, the (α, β) space describes the change of 2-D flow patterns as a progression around the unit circle. Traditionally, phase-portraits are classified on $p - q$ charts [3]. Though other mappings are possible, the (α, β) representation provides a logical grouping and progression from one critical point to another thereby being attractive for comparing critical points. For example, vortical behavior is grouped together ($\beta < 0$) in the lower half of the unit circle. From dynamics it is known that a center pattern ($\beta = -1$) is structurally unstable (non-hyperbolic) and therefore is represented as a single point in this space. Any slight perturbation (typically introduced from non-linearities in the system) can change the pattern to an attracting or repelling focus ($\beta \approx 1$). Therefore the distance between these focii can be small as found for nearly conservative systems. The Earth mover's distance (EMD) was used as the matching algorithm to minimize the cost of work between the fields. The work in this case was merely the Euclidean distance between two (α, β) points. However, the connectivity between critical points was not used. This could be problematic since two vector fields with similar critical points but significantly different connections led to similar EMD values and therefore were false positives. For a topologically accurate representation, it is necessary to include not only the number and type of critical points but the set of connecting streamlines.

Two methods are presented that define metrics for comparing the number, type, and connections between vector fields. The first technique is based on graph theory where the vector field is represented using an attributed, relational graph (ARG) [4]. The second method is based on EMD with a redefined feature distribution that captures the graphical structure of the vector field. The trade-off between the methods is accuracy versus speed. The paper is concluded with an application demonstrating the advantages over the original method.

3 Representation of Vector Fields using Attributed, Relational Graphs

An attributed relational graph is composed of a set of property based nodes and the relations between the nodes represented by edges or connections. In the case of vector field topology, the nodes are the critical points with (α, β) as the characteristic property and the connecting streamlines are the edges that relate the critical points. To compare two attributed graphs, G_i and G_j, a similarity or distance measure, $D(G_i, G_j)$ is defined between them. The measure which can be thought of as a cost to transform G_i to G_j involves both the cost of transforming the properties of the nodes and the relations of the graph. A desirable property for $D(G_i, G_j)$ is that it be a metric [4].

The process of transforming the properties of the node (node-matching) can be interpreted as the total cost, c_n, required to transform a set of nodes $\{p\}$ of G_i to the set of nodes $\{q\}$ of G_j. Therefore, given a particular node-pairing, x, where $f_n(p_i, q_j)$ is the cost of transforming a pair of nodes (p_i, q_j), the total cost can be written as $c_n = \sum f_n(p_i, q_j)$ where the summation is over all node pairs. For vector fields, the cost of node matching is merely the Euclidean distance between (α, β) points [1]. If two vector fields contain an unequal number of critical points, normal points corresponding to $(\alpha, \beta) = (0, 0)$ are added. Normal points correspond to node insertion and deletion and have a uniform cost of unity since the distance from a normal point to any point on the unit circle is 1.

The relation between two graphs is equivalent to graph isomorphism which quantifies how close one graph structure is to another. In order to equate two graphs, node insertion and deletion as well as edge insertion or deletion are required. The edge relations between critical points for a vector field are obtained by numerically integrating the streamlines connecting the critical points [5]. The edges can then be related by an adjacency matrix, A_{ij} [4]. This nxn matrix where n is the number of nodes (critical points) is assigned a value of 1 if critical point i is connected to critical point j, otherwise a 0 is assigned. Note that the adjacency matrix can represent only one connection between two nodes and has implications to be discussed. To compare graphs G_i and G_j, the corresponding adjacency matrix is compared component wise. The node-matching procedure described above guarantees that the size of all adjacency matrices are the same since normal points are added to the flow field. Since the graph is undirected (i.e. node i is connected to node j, and vice versa), the adjacency matrix is symmetric and only half the matrix need be considered). The overall cost involved to convert vector field G_i with a configuration x and adjacency matrix A to another vector field, G_j with matrix A' is the sum of the relational-matching and node-matching components:

$$D_S(x) = \sum_{i=1}^{n} \sum_{j=i}^{n} |A_{ij} - A'_{ij}| + \sum_{i,j} \sqrt{(\alpha_i - \alpha_j)^2 + (\beta_i - \beta_j)^2} \quad (1)$$

And the distance measure is defined as the minimum over all possible node pairings and the corresponding permutation of the adjacency matrix,

$$D(G_i, G_j) = \overset{min}{x} \; D_S(x) \quad (2)$$

3.1 Comments on the ARG Technique

Since all permutations of the adjacency matrix need to be calculated, the overall complexity is $O(n!\ n^2)$ [5] which is unacceptable for large problems. However, significant research in the area of graph matching exists and is currently being investigated [6]. The adjacency matrix described is binary and does not take into account graphs with parallel edges. In the case of a saddle point, 2 streamlines may connect to a single critical point. However, non-binary matrices can be used and is being investigated as well [7].

4 Extending EMD to Graph Matching

As forementioned, an immediate disadvantage to the current implementation of ARGs is the time required for comparison and therefore is an area of current investigation. However, a readily extendible solution is to modify the signature defined in [1] to account for the connections between critical points. The EMD method in the worst case requires 2^n iterations, but in practice has been shown to be solved in polynomial time (cubic) [8].

In the previous work, a feature distribution for a vector field is the set of (α, β) values associated with the critical points [1]. A new feature distribution is defined to be a set of (α, β) values and the (α, β) values of the immediate set of connections. This approximates a graph structure with n tree structures. The decomposition is not unique and therefore cases can be contrived such that two different graphs can be decomposed into the same tree structures [5]. However, given the relative position and types of critical points, there are a limited number of ways that streamlines can join the points, and as will be shown in the application section, in practice this method is comparable to ARG.

The work required to convert one vector field into another is defined to be the *Set Work*.

Definition 1 (Set Work) *For two vector fields with feature*

distribution

$$\{ \{(\alpha_1,\beta_1),(\alpha_1^{(1)},\beta_1^{(1)}),\ldots,(\alpha_1^{(n_1)},\beta_1^{(n_1)})\},$$
$$\{(\alpha_2,\beta_2),(\alpha_2^{(1)},\beta_2^{(1)}),\ldots,(\alpha_2^{(n_2)},\beta_2^{(n_2)})\},$$
$$\ldots\{(\alpha_n,\beta_n),(\alpha_n^{(1)},\beta_n^{(1)}),\ldots,(\alpha_n^{(n_n)},\beta_n^{(n_n)})\} \}$$

and

$$\{ \{(\hat{\alpha}_1,\hat{\beta}_1),(\hat{\alpha}_1^{(1)},\hat{\beta}_1^{(1)}),\ldots,(\hat{\alpha}_1^{(n_1)},\hat{\beta}_1^{(n_1)})\},$$
$$\{(\hat{\alpha}_2,\hat{\beta}_2),(\hat{\alpha}_2^{(1)},\hat{\beta}_2^{(1)}),\ldots,(\hat{\alpha}_2^{(n_2)},\hat{\beta}_2^{(n_2)})\},$$
$$\ldots\{(\hat{\alpha}_n,\hat{\beta}_n,(\hat{\alpha}_n^{(1)},\hat{\beta}_n^{(1)}),\ldots,(\hat{\alpha}_n^{(n_n)},\hat{\beta}_n^{(n_n)})\} \}$$

the amount of work necessary for transforming one vector field into another is defined as:

$$Work_{Set} = \sum_{i=1}^{n}[\, Work_{ES}(\{(\alpha_i,\beta_i)\},\{(\hat{\alpha}_i,\hat{\beta}_i)\})$$
$$+ min(Work_{ES})(\{(\alpha_i^{(1)},\beta_i^{(1)}),\ldots,(\alpha_i^{(n_i)},\beta_i^{(n_i)})\},\ \ldots,$$
$$\{(\hat{\alpha}_i^{(1)},\hat{\beta}_i^{(1)}),\ldots,(\hat{\alpha}_i^{(n_i)},\hat{\beta}_i^{(n_i)})\}\,)\,]$$

where $Work_{ES}$ is the Elemental Set Work.

The $min(Work_{ES})$ is the minimum amount of elemental work. This can be found using EMD. The Elemental Set Work remains the same and is defined in [1]. Therefore, the distance measure is defined as the minimum of the set work and is computed using EMD. To summarize, EMD is run at two levels. The first is at a global level where EMD minimizes the set work required to convert one vector field into another. The set work which converts one critical point into another requires the use of EMD to minimize the work required to convert one critical point's children into another critical points's children.

4.1 Comments on Extending EMD Method

As mentioned previously, the representation of a graph through a tree structure is not necessarily unique and therefore can expect false positives. However, it may be possible to determine up front whether a particular graph has a unique tree structure representation. In this case, the method works nearly as well as ARG. The results will not be the same, since a tree structure does not distinguish clearly between the cost of node-matching and graph-matching as it does with ARG. In the limiting case, when no connections are present, the extended method produces the same results as the original EMD algorithm. Therefore, a lower limit can be established. The extended method is a metric since the *Set Work* is a composition of two metrics. Rubner has proven that EMD is a metric if the distance measure is a metric. [8].

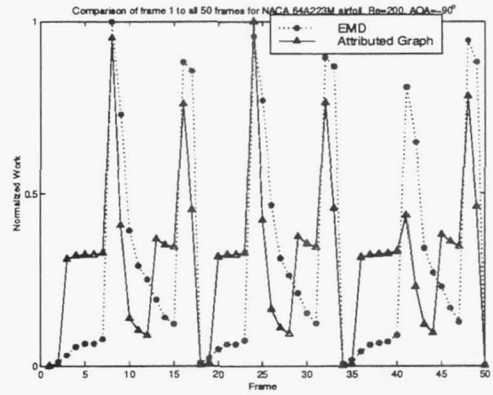


Figure 1: Attributed, Relational Graph compared with Original EMD method. Note differences in frames 3-7,13 and 41.

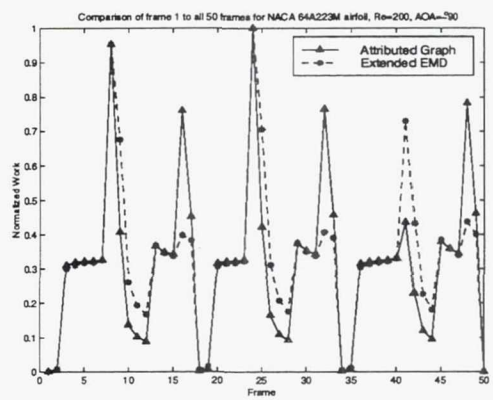


Figure 2: Attributed, Relational Graph compared with Extended EMD method. Note that the trends agree.

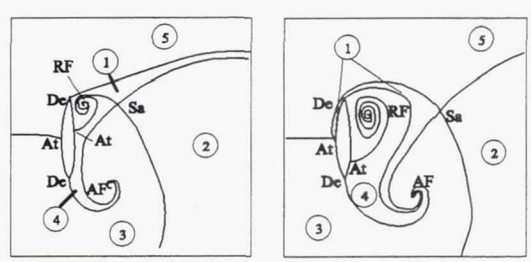


Figure 3: Graph comparison of two vector fields. Region 1 in Frame 4 extends to the upstream region of the airfoil.

5 Applications

To demonstrate the effectiveness of ARG and extended EMD, a comparison is made to the original EMD algorithm on the test case of a flow past a 2-D airfoil at -90° angle of attack. Rogers and Kwak computed this flow using an incompressible, time accurate Navier-Stokes code at Re=200 [9]. Fifty time steps were computed capturing several periodic vortex shedding cycles. Figure 1 plots the EMD of frame 1 versus the remaining 49 frames. The plots have been normalized by the largest EMD values so the techniques may be compared. Intuitively, Figure 1 represents how much vector field 1 differs from the remaining 49 fields. Both methods capture the periodicity within the flow indicated by the valleys-every 16 to 17 frames. However, the ARG algorithm differs from the original EMD in frames 3-7. From the topological illustrations in Figure 3, we see that contrary to EMD's computation frames 1 and 4 are not equivalent. Region 1 in frame 4 extends to the upstream region of the airfoil. Also a new region is present starting in frame 13. Note that the original EMD method used only simple critical points and did not include the wall attachment/detachment points. In order to preserve the graphical structure, the separation points have been represented by normal points $(\alpha, \beta) = (0, 0)$, but will be properly classified in future papers. The number and type of attachment/detachment points do not change with time (always 2 attachment and 2 detachment points) and so the methods are comparable. The cycle is consistently completed every 16 to 17 frames. However, one marked difference occurs in frame 41 which has an energy level lower than its counter parts in frames 8 and 25. This is due to frame 41 missing a saddle/repelling focus pair, and therefore the graph structure is closer to frame 1. Hence the graph method has indicated a possible problematic area in the solver.

Finally, Figure 2 compares the ARG method with the extended EMD method. Immediately, it can be seen that both methods are in agreement with one another. The amplitudes are not expected to agree since each algorithm weighs the significance of the nodes differently.

6 Conclusions

We have demonstrated the effectiveness of including connectivity information in the representation of vector field comparisons. By further considering the connecting lines forming the separatrices of the vector field, we are able to improve the quantitative measure between fields. For the airfoil data, we have shown that both the attributed graph method, and the extended EMD method can be used as an effective investigative and diagnostic tool. The extended EMD method is considerably faster than the brute force ARG method and in practice should be considered as an effective first pass tool.

ACKNOWLEDGEMENT

The authors are most indebted to Yossi Rubner for providing EMD and MDS codes. Kerstin Kling is grateful to the Ernest-Solvay Foundation for the generous financial support during her stay at Stanford University.

References

[1] Y. Lavin, R. Batra, and L. Hesselink, "Feature comparisons of vector fields using earth mover's distance," in *Proc. IEEE/ACM Visualization '98*, pp. 413–415, North Carolina, October 1998.

[2] Q. Shen, A. Pang, and S. Uselton, "Data level comparison of wind tunnel and computational fluid dynamics data," in *Proc. IEEE Visualization '97*, pp. 67–74, CS Press, Los Alamitos, CA., 1997.

[3] W. E. Boyce and R. C. DiPrima, *Elemetrary Differential Equations and Boundary Value Problems*. John Wiley & Sons, fourth ed., 1986.

[4] R. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons Inc., 1992.

[5] K. Kling, "Topology based quantitative comparison between vector fields," Master's thesis, University of Hannover, 1999. Diplomarbeit.

[6] B. Messmer, *Efficient Graph Matching Algorithms for Preprocessed Model Graphs*. PhD thesis, University of Bern, 1996.

[7] R. Busacker and T. Saaty, *Finite graphs and networks; an introduction with applications*. McGraw-Hill Book Company, 1965.

[8] Y. Rubner and C. Tomasi, "The earth mover's distance as a metric for image retrieval," Tech. Rep. STAN-CS-TN-98-86, Department of Computer Science, Stanford University, September 1998.

[9] S. Rogers and D. Kwak, "An upwind differencing scheme for the time-accurate incompressible navier-stokes equations," in *Proceedings of the AIAA 6th Applied Aerodynamics Conference*, pp. 492–502, American Institute of Aeronautics and Astronautics, 1988.

The Topology of Three-Dimensional Symmetric Tensor Fields

Yingmei Lavin
Department of Physics
Stanford University
Stanford, CA 94305

Yuval Levy
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305-4035

Lambertus Hesselink
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-4035

1 Abstract

We study the topology of 3-D symmetric tensor fields. The goal is to represent their complex structure by a simple set of carefully chosen points and lines analogous to vector field topology. The basic constituents of tensor topology are the degenerate points, or points where eigenvalues are equal to each other. First, we introduce a new method for locating 3-D degenerate points. We then extract the topological skeletons of the eigenvector fields and use them for a compact, comprehensive description of the tensor field. Finally, we demonstrate the use of tensor field topology for the interpretation of the two-force Boussinesq problem.

2 Introduction

Second-order tensor fields have applications in many areas of physics, such as general relativity, fluid flows and mechanical properties of solids. The wealth of multivariate information in tensor fields makes them more complex and abstract than scalar and vector fields. Visualization provides a means to gain new insights from these rich data sets.

The most natural way to visualize a symmetric 3-D tensor field is through its eigensystem, i.e., eigenvalues and eigenvectors. A continuous representation of the tensor field is obtained by tracing the trajectories of its eigenvectors. These trajectories are called hyperstreamlines [1, 2]. The difficulty with such an approach is how to capture the structure of the 3-D domain while limiting the number of hyperstreamlines to a minimum in order to avoid visual clutter. The problem can be significantly simplified by taking a topological approach, similar to the one used in visualizing vector fields [3]. Degenerate points, defined as points where eigenvalues are equal to each other, are the basic singularities underlying the topology of tensor fields. Eigenvectors never cross each other except at degenerate points. In the past, research has been conducted in the area of two-dimensional tensor fields [1, 2]. We live, however, in a three-dimensional world, and therefore,it is important for us to understand the underlying physics of this world. In this paper, we describe a new method for locating degenerate points along with the conditions for classifying them in three-dimensional space. We also discuss some topological features of three-dimensional tensor fields, and interpret topological patterns in terms of physical properties.

3 Theoretical Background

3.1 Definitions

Definition 1 (Second-Order Tensor Field) *Let $\mathcal{L}(X,Y)$ be the set of all the linear transformations of the vector space X into the vector space Y, and let E be an open subset of $\mathbf{R}^n$. A second-order tensor field $\mathbf{T}(\vec{x})$ defined across E is a mapping $\mathbf{T} : E \rightarrow \mathcal{L}(\mathbf{R}^m, \mathbf{R}^m)$ that associates to each vector $\vec{x}$ of E a linear transformation of $\mathbf{R}^m$ into itself. If $\mathbf{R}^m$ is referenced by a Cartesian coordinate system, $\mathbf{T}(\vec{x})$ can be represented by m^2 Cartesian components $T_{ij}(\vec{x})$, $i,j = 1,\ldots,m$, that transform according to*

$$T'_{ij} = \sum_{p,q=1}^{m} \beta_{ip}\beta_{jq}T_{pq} \tag{1}$$

under an orthonormal transformation $\beta = \{\beta_{ij}\}$ of the coordinate axes. [2]

In a Cartesian coordinate system, a 3-D tensor field takes the following form:

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} T_{11}(x,y,z) & T_{12}(x,y,z) & T_{13}(x,y,z) \\ T_{21}(x,y,z) & T_{22}(x,y,z) & T_{23}(x,y,z) \\ T_{31}(x,y,z) & T_{32}(x,y,z) & T_{33}(x,y,z) \end{pmatrix} \tag{2}$$

Definition 2 *(Hyperstreamline) A geometric primitive of finite size sweeps along the longitudinal eigenvector field, $\vec{v}_l$, while stretching in the transverse plane under the combined action of the two transverse eigenvectors, $\vec{v}_{t1}$ and $\vec{v}_{t2}$. Hyperstreamlines are surfaces that envelop the stretched primitives along the trajectories. We refer to hyperstreamlines as "major", "medium" or "minor" depending on the corresponding longitudinal eigenvector field that defines their trajectories and color hyperstreamlines by means of a user-defined function of the three eigenvalues, usually the amplitude of the longitudinal eigenvalue. [4]*

Definition 3 *A degenerate point of a tensor field $\mathbf{T} : E \rightarrow \mathcal{L}(\mathbf{R}^m, \mathbf{R}^m)$, where E is an open subset of $\mathbf{R}^m$, is a point $\vec{x}_0 \in E$ where at least two of the m eigenvalues of $\mathbf{T}$ are equal to each other. [4]*

3.2 Locating Degenerate Points

A three-dimensional symmetric tensor field (Equation (2)) has 6 independent variables, therefore various types of degenerate points may exist. These types correspond to the following conditions:

$$\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) > \lambda_3(\vec{x}_0) \tag{3}$$

$$\lambda_1(\vec{x}_0) > \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0) \tag{4}$$

$$\lambda_1(\vec{x}_0) = \lambda_2(\vec{x}_0) = \lambda_3(\vec{x}_0) \tag{5}$$

The characteristic equation of a 3-D symmetric tensor can be expressed in the following form

$$\begin{aligned} A(\lambda) &= \begin{vmatrix} T_{11}-\lambda & T_{12} & T_{13} \\ T_{12} & T_{22}-\lambda & T_{23} \\ T_{13} & T_{23} & T_{33}-\lambda \end{vmatrix} \\ &= -\lambda^3 + a\lambda^2 + b\lambda + c \end{aligned} \tag{6}$$

where a, b and c are composed of the 6 independent tensor components. The condition for the existence of a degenerate point is that both $A(\lambda(\vec{x}_0))$ and its derivative $\frac{dA(\lambda(\vec{x}_0))}{d\lambda}$ are zero.

$$\begin{cases} A(\lambda(\vec{x}_0)) = -\lambda^3 + a\lambda^2 + b\lambda + c = 0 \\ \frac{dA(\lambda(\vec{x}_0))}{d\lambda} = -3\lambda^2 + 2a\lambda + b = 0 \end{cases} \tag{7}$$

As a result, we obtain the following conditions corresponding to Equations (3, 4, and 5) respectively:

$$B_1(x,y,z) = \frac{2a^3 + 9ab + 2d^{3/2}}{27} + c = 0 \tag{8}$$

$$B_2(x,y,z) = \frac{2a^3 + 9ab - 2d^{3/2}}{27} + c = 0 \tag{9}$$

$$B_3(x,y,z) = a^2 + 3b = 0 \tag{10}$$

From the expressions for B_1, B_2 and B_3, we determine that: $B_1(x,y,z) = 0$ is a maximum for B_1, $B_2(x,y,z) = 0$ is a minimum for B_2 and $B_3(x,y,z) = 0$ is a maximum for B_3.

Now the problem is to find extrema in a 3D continuous field from the discrete experimental data sets. On a 3-D discrete mesh, the search for the various extrema is conducted by processing one grid cell at a time for each spatial function.

This method can successfully locate the points of triple degeneracy. It is especially useful when extended to locate points of double degeneracy where the local tensor appears in the diagonal form only when transformed into its eigenvector space.

3.3 Separating Surfaces

For second order tensor fields, in most cases, the eigenvector fields in the vicinity of a degenerate point can be described in terms of three types of angular sectors: hyperbolic, parabolic and elliptic sectors. It can be proved that in a 2-D tensor field, at a simple degenerate point, there are only one or three hyperbolic sectors, and no elliptic sectors [2]. Correspondingly, we call the degenerate point a wedge point when it has only one hyperbolic sector and possibly one parabolic sector or a trisector point when it has three hyperbolic sectors [2].

The classification of degenerate points in 2-D tensor fields [2, 5] can then be extended to 3-D tensor fields. The building blocks are the fundamental elements as defined for 2-D [2]. However, the separating surfaces in 3-D tensor fields have a general structure as they could appear at various angles. Each of the surfaces is characterized by patterns similar to those of hyperbolic or parabolic sectors and is bounded by hyperstreamlines that are emanating from or terminated at the degenerate point. Consequently, a point of triple degeneracy can be classified by the number and type of separating surfaces surrounding it.

In Figure 1 we show the eigenvector patterns in the vicinity of a point of triple degeneracy with 4 bounding hyperstreamlines. These hyperstreamlines form 6 hyperbolic separating surfaces. Figure 2 shows a point of triple degeneracy with only 3 bounding hyperstreamlines which form 2 hyperbolic separating surfaces and one parabolic surface.

The trajectories on the surfaces are locally 2-D, while off the surfaces they are fully 3-D and are determined by their closest surface.

4 Topology of 3-D Tensor Fields

We choose the elastic stress tensor induced by two compressive forces on the top of a semi-infinite

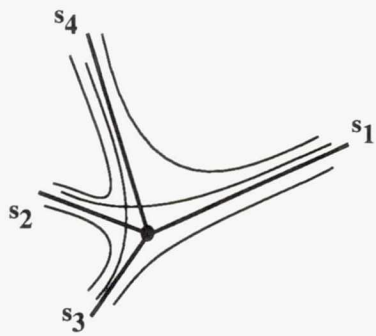


Figure 1: A point of triple degeneracy with 6 hyperbolic separating surfaces.

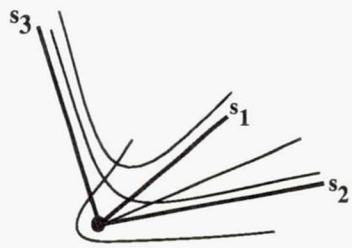


Figure 2: A point of triple degeneracy with 2 hyperbolic separating surfaces and one parabolic surface.

plane [6] to illustrate the advantages of using topological skeletons in visualizing 3-D tensor fields. In principle, hyperstreamline trajectories of the stress tensor show the transmission of forces inside the material. Figure 3 shows two hyperstreamlines corresponding to the most compressive eigen-direction, the minor eigenvector $\overline{v}_3$. The two forces, indicated by the arrows, act on the surface at $\mathbf{P}_1 = (0.5, 0.0, -1.05)$ and $\mathbf{P}_2 = (-0.5, 0.0, -1.05)$ in the $+z$ direction (downward). The domain of interest (described by the bounding frame) extends between $(-1.0, -1.0, -1.114367)$ and $(1.0, 1.0, 0.0)$ so it includes the key features of the stress tensor field, i.e., the degenerate points. It is assumed that the region where $z < -1.05$ is in tension and that no stresses are transferred across the plane $z = -1.05$. The color of the hyperstreamlines encodes the magnitude of the most compressive eigenvalue, λ_3, while their cross section encodes the magnitude and direction of the transverse eigenvectors. The hyperstreamlines converge toward regions of high stresses where the forces are applied. Note the sharp change in color and cross section size of the hyperstreamlines as they approach the acting points of the forces.

Analysis reveals that the tensor field contains two points of triple degeneracy and that these points reside on the surface of the semi-infinite plane. Moreover, the eigenvalues at these points (the location of which is given by: $\mathbf{D}_1 = (0.0, 0.5, -1.05)$, $\mathbf{D}_2 = (0.0, -0.5, -1.05)$) are equal to zero. This means that

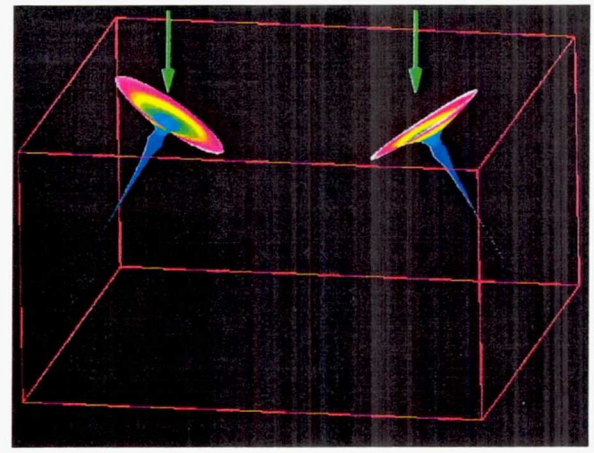


Figure 3: Stress tensor induced by two compressive forces; minor hyperstreamlines

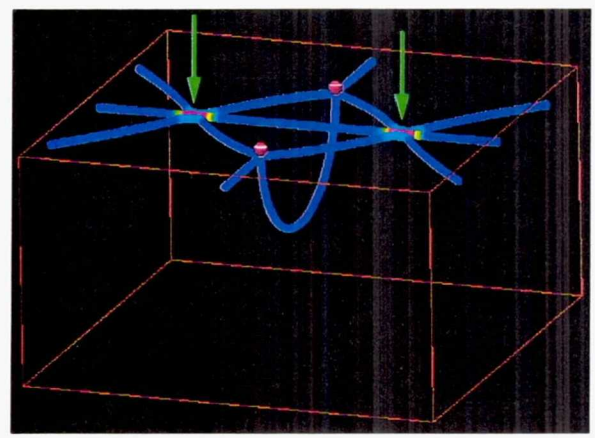


Figure 4: Stress tensor induced by two compressive forces; major hyperstreamlines

these points are stress free, a fact that can be verified by an examination of the stress equations. We have therefore acquired physical insight into the stress tensor field just by examining a basic topological feature, a point of triple degeneracy.

Figure 4 shows hyperstreamlines that are obtained by tracing the major eigenvector field. The location and direction of the forces are indicated by the arrows and the location of the points of triple degeneracy are marked by spheres. The hyperstreamlines are presented with a constant cross section to avoid visual clutter resulting from the high eigenvalues in the vicinity of the points of the acting forces. They are, however, still color encoded by the major eigenvalue. Each of the 2 degenerate points has 4 bounding hyperstreamlines(separatrices), three of which lie on the surface $z = -1.05$ in a trisector pattern and the forth, which is pointing in the $+z$ direction, connects the points of triple degeneracy, and delineates one of the two symme-

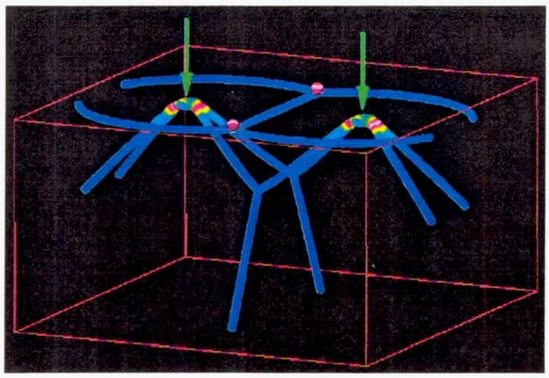


Figure 5: Stress tensor induced by two compressive forces; minor hyperstreamlines

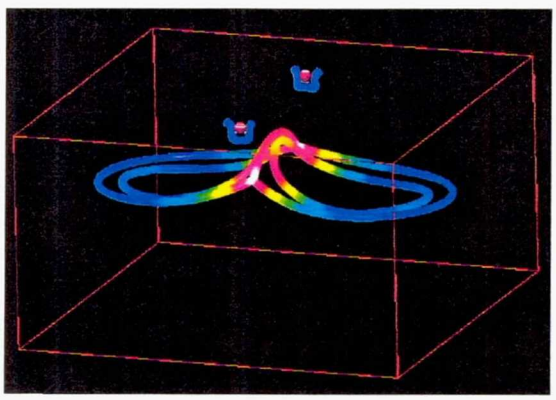


Figure 6: Stress tensor induced by two compressive forces; medium hyperstreamlines

try planes (the other goes through the points of action of the forces).

To further clarify the tensor topology, the skeletons of the minor and medium hyperstreamlines are presented in Figures 5 and 6 respectively. We can see from Figure 5 that the minor hyperstreamlines form a trisector-point like pattern in the vicinity of the points of triple degeneracy. They also indicate that a locus of points of double degeneracy ($\lambda_2 = \lambda_3$) connects the points of triple degeneracy. This is evident from the two trisector points that lie in the symmetry planes just below the points of triple degeneracy. The existence of the line of double degeneracy is further verified by noting the two points of double degeneracy in the skeleton of the medium hyperstreamlines (Figure 6).

5 Conclusions

In this paper, we applied novel methods to determine the topology of tensor data sets, and made use of advanced representations to determine the significance of degenerate points and topological skeletons in terms of physical features.

By extracting the geometric structure of tensor data, we produce simple and austere depictions that allow observers to infer the behavior of any hyperstreamlines in the field. It enables important elements of 3-D stress distribution to be envisaged without visual clutter.

Degenerate points represent the singularities of the tensor field. In the 3-D elastic stress tensor case we were able to identify points of zero stresses with triple degenerate points and to illustrate transmission of forces inside the material.

Acknowledgment

We are most indebted to Rajesh Batra from Stanford University for his critical comments and some of his visualization software. The authors are supported by NASA under contract NAG 2-911 which includes support from the NASA Ames Numerical Aerodynamics Simulation Program and the NASA Ames Fluid Dynamics Division, and also by NSF under grant ECS-9215145.

References

[1] T. Delmarcelle and L. Hesselink, "Visualization of second-order tensor fields and matrix data," in *Proc. IEEE Visualization '92*, pp. 316–323, CS Press, Los Alamitos, CA., 1992.

[2] T. Delmarcelle, *The Visualization of Second-Order Tensor Fields*. PhD thesis, Stanford University, 1994.

[3] J. L. Helman and L. Hesselink, "Visualization of vector field topology in fluid flows," *IEEE Computer Graphics and Applications*, vol. 11, no. 3, pp. 36–46, 1991.

[4] T. Delmarcelle and L. Hesselink, "Visualizing second-order tensor fields with hyperstreamlines," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25–33, 1993.

[5] T. Delmarcelle and L. Hesselink, "A unified framework for flow visualization," in *Computer Visualization* (R. Gallagher, ed.), ch. 5, CRC Press, 1994.

[6] H. M. Westergaard, *Thoery of Elasticity and Plasticity*. Cambridge, Harvard University Press, 1952.

The Topology of Symmetric Tensor Fields

Yingmei Lavin* Rajesh Batra† and Lambertus Hesselink‡

Stanford University, Stanford, California 94305-4035

Yuval Levy§

Technion, Israel Institute of Technology, Haifa, Israel 32000

Abstract

Combinatorial topology, also known as "rubber sheet geometry", has extensive applications in geometry and analysis, many of which result from connections with the theory of differential equations. A link between topology and differential equations is vector fields. Recent developments in scientific visualization have shown that vector fields also play an important role in the analysis of second-order tensor fields. A second-order tensor field can be transformed into its eigensystem, namely, eigenvalues and their associated eigenvectors without loss of information content. Eigenvectors behave in a similar fashion to ordinary vectors with even simpler topological structures due to their sign indeterminacy. Incorporating information about eigenvectors and eigenvalues in a display technique known as hyperstreamlines[1] reveals the structure of a tensor field. To simplify an often complex tensor field and to capture its important features, the tensor is decomposed into an isotropic tensor and a deviator. A tensor field and its deviator share the same set of eigenvectors, and therefore they have a similar topological structure. A deviator determines the properties of a tensor field, while the isotropic part provides a uniform bias. Degenerate points are basic constituents of tensor fields. In 2-D tensor fields, there are only two types of degenerate points; while in 3-D, the degenerate points can be characterized in

*Graduate Student, Physics Department

†Graduate Student, Aeronautics and Astronautics Department

‡Professor, Electrical Engineering and Aeronautics and Astronautics Departments, Member AIAA.

§Lecturer, Faculty of Aerospace Engineering, Member AIAA.

Copyright © 1997 by Authors. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

a Q'-R' plane. Compressible and incompressible flows share similar topological features due to the similarity of their deviators. In the case of the deformation tensor, the singularities of its deviator represent the area of the vortex core in the field. In turbulent flows, the similarities and differences of the topology of the deformation and the Reynolds stress tensors reveal that the basic eddie-viscosity assumptions have their validity in turbulence modeling under certain conditions.

Introduction

Combinatorial topology is a branch of geometry. It studies the properties of figures that endure when the figures are subjected to continuous transformations. A topological property of a figure is a property possessed alike by the figure and all its topological equivalents. In combinatorial topology, complicated figures are constructed from simple ones and their properties are deduced from the simple figures.[2]

Combinatorial topology has vast applications to solving systems of differential equations. It may seem surprising that such superficially different subjects as topology and differential equations could be related, but research has shown that a link between these two subjects is the concept of a vector field. Vector fields have many important applications. The force fields arising from gravitation and electromagnetism; the velocity vectors of a fluid motion, such as the atmosphere; and gradients, such as the pressure gradient on a weather map or the height gradient of a relief chart are all elements of vector fields. Recent developments in scientific visualization have shown that vector fields and their topological structures also play a very important role in analyzing second-order tensor fields.

Second-order tensors are fully represented by their eigenvectors and associated eigenvalues.

$$\mathbf{T}\vec{e_i} = \lambda_i \vec{e_i} \tag{1}$$

Where λ_i and $\vec{e_i}$ $(i = 1, 2, 3)$ are eigenvalues and eigenvectors of the tensor $\mathbf{T}$, respectively. The λ_i's represent all the amplitude information while the $\vec{e_i}$'s represent all the directional information of $\mathbf{T}$. Visualizing a tensor field is equivalent to visualizing its eigenvector fields. However, unlike a vector field, Eigenvectors are vectors with sign indeterminacy. This remarkable feature distinguishes eigenvector fields from ordinary vector fields and makes their topological features even simpler. The basic constituents of tensor topology are degenerate points where at least two of the eigenvalues are equal to each other. They play a role similar to critical points in vector fields. As with vector fields there are numerous tensor quantities that are of interest to analyze and

visualize. In fluid mechanics, the Reynolds stress and the strain-rate (deformation) tensors are examples of tensors of high interest. This is in part due to the fact that eddie viscosity turbulence models are based on the assumption that the Reynolds stress tensor can be modeled as a function of the deformation tensor. A comparison of the topology of the two tensors would provide a means for validating the basic assumptions. Another aspect of the deformation tensor is the alignment of its medium eigenvector with the vorticity vector and scalar gradients.[3] [4] It is believed that a stationary flow near the vortex core under goes rigid body motion implying that the components of the deformation tensor are zero. Computation and visualization of the phenomena is still an area of research and debate. Other tensors commonly used in the area of fluid mechanics are the velocity gradient, stress tensor, and the momentum-flux-density tensor.

Theoretical Background

Critical Points and Flow Field Topology

A critical point is a point in the flow field where all three velocity components are zero and the streamline slope is indeterminate. The number and nature of critical points in a vector field remain unchanged under a continuous transformation. These are topological properties of the system of differential equations.

Historically, critical point theory has been used primarily to examine the solution trajectories of ordinary differential equations.(see Kaplan ,[5] Pontryagin,[6] Andronov,[7] and Minorsky[8]) In a reference frame where a critical point is located at the origin, the motion in the flow is described by the leading terms of a Taylor expansion for the velocity field. The streamlines are, therefore, defined by the solution trajectories of three linear, coupled, first-order ordinary differential equations. The relationship between the properties of this 3×3 Jacobian matrix and the geometry of the solution trajectories is not trivial.

A critical point can be classified according to the eigenvalues of its Jacobian matrix.[9,10] A positive or negative real part of an eigenvalue indicates a repelling or attracting nature, respectively. The imaginary part denotes circulation about the point. Accordingly, a critical point can be classified as an *attracting node*, a *repelling node*, an *attracting focus*, a *repelling focus*, a *center* or a *saddle*.

For a complex flow field, the eigenvector planes are not always easily identified and located and it becomes necessary to use an analysis based on three invariants P,Q and R of a 3×3 Jacobian matrix.[11] All possible linear local flow trajectories of a moving continuum for both compressible and incompressible fluids can

be completely categorized in the space of P,Q and R. A set of surfaces can be defined in this space which defines boundaries between topologically distinct flow patterns and serves as a guide for identifying critical points.[11]

Tensor Field Analysis

A tensor field in the real world is often very complex and difficult to analyze as a whole. Therefore, reducing the field to a simpler form is desirable.

Decomposition of a Tensor Field

Any given tensor can be decomposed into a sum of a **deviator D** and an **isotropic tensor U**:

$$\mathbf{T} = \mathbf{D} + \mathbf{U} \tag{2}$$

Definition 1 (Deviator) *A tensor is a deviator* $\mathbf{D}$ *iff it is trace free, i.e. Trace* $(\mathbf{D}) = 0$,

Definition 2 (Isotropic Tensor)
A tensor is isotropic iff $U_{ij} = \nu\delta_{ij}$, *where* ν *is a stretch factor.*

Definition 3 (Degenerate Point) *A degenerate point of a tensor field* $\mathbf{T} : E \rightarrow \mathcal{L}(\mathbf{R}^m, \mathbf{R}^m)$, *where* E *is an open subset of* $\mathbf{R}^m$, *is a point* $\vec{x}_0 \in E$ *where at least two of the* m *eigenvalues of* $\mathbf{T}$ *are equal to each other.*

Definition 4 (Singular Point) *A singular point in a tensor field is a point where all eigenvalues of a tensor vanish, in mathematical representation, it is a zero matrix.*

At any location in a tensor field, an **isotropic tensor** behaves the same in every direction, in other words, it is isotropic throughout the whole field. The topology of such a tensor is fairly simple and easy to deduce.

A **deviator**, in contrast to an **isotropic tensor**, has a different behavior in all 3 principal directions except at a singular point where all of its components are zero. A general tensor and its **deviator** have the same set of eigenvectors; and degenerate points of a general tensor and singular points of its **deviator** occur at the same locations.[12] This means that the topology of a tensor field is identical to the topology of its deviator. A real tensor field is a **deviator** superimposed onto an **isotropic tensor**. By subtracting the contribution of the **isotropic tensor** from the tensor field, the **deviator** becomes dominant. This allows a clear depiction of the topology and the fluctuations of the field without the disturbance of the sometimes dominant isotropic contribution. In the following classification of a 3-D tensor field topology, analysis of the **deviator** also provides a mathematical simplification that reduces the otherwise 3-D P-Q-R

space into a 2-D Q-R plane.

Hyperstreamlines

Similar to streamlines in a flow field, hyperstreamlines are integral curves satisfying:

$$\frac{d\overline{x}}{ds} = \overline{v}\left(\overline{x}\right) \tag{3}$$

Where s is a parameter measuring distance along the path. Unlike the velocity field, $\overline{v}$ represents one of the eigenvectors of a tensor field $\mathbf{T}\left(\overline{x}\right)$ obtained from:

$$\mathbf{T}\left(\overline{x}_0\right)\overline{v}\left(\overline{x}_0\right) = \lambda\left(\overline{x}_0\right)\overline{v}\left(\overline{x}_0\right) \tag{4}$$

at each location $\overline{x}_0$.

A hyperstreamline in 3-D is a trajectory that traces along the longitudinal eigenvector field while stretching in the transverse plane under the combined action of the two transverse eigenvectors. We refer to hyperstreamlines as "major", "medium" and "minor" depending on the corresponding longitudinal eigenvector field that defines its trajectory.

Topology of Tensor Fields

The topology of a tensor field $\mathbf{T}(\overline{x})$ is the topology of its eigenvector fields $\overline{v}_i(\overline{x})$.[13] Similar to critical points, degenerate points are the basic constituents underlying the topology of tensor fields. Degeneracy occurs at points where at least two eigenvalues are equal to each other.

2-D Tensor Fields

In the case of two-dimensional tensor fields, there are only two eigenvalues λ_1 and λ_2, and $\overline{x}_0$ is a degenerate point iff $\lambda_1\left(\overline{x}_0\right) = \lambda_2\left(\overline{x}_0\right)$.

Similar to critical points in vector fields, tensor fields have different types of degenerate points that correspond to different patterns in their neighborhoods. These patterns are determined by the tensor gradients at the degenerate points.[14–16]

Consider the partial derivatives

$$\begin{array}{ll} a = \frac{1}{2}\frac{\partial(T_{11}-T_{22})}{\partial x} & b = \frac{1}{2}\frac{\partial(T_{11}-T_{22})}{\partial y} \\ c = \frac{\partial T_{12}}{\partial x} & d = \frac{\partial T_{12}}{\partial y} \end{array} \tag{5}$$

evaluated at the degenerate point $\overline{x}_0$. In the vicinity of $\overline{x}_0$, the expansion of the tensor components to first-order is as follows:

$$\begin{cases} \frac{T_{11}-T_{22}}{2} \approx a\Delta x + b\Delta y \\ T_{12} \approx c\Delta x + d\Delta y \end{cases} \tag{6}$$

where $(\Delta x, \Delta y)$ are small displacements from $\overline{x}_0$. An important quantity for the characterization of degenerate points is the quantity

$$\delta = ad - bc \tag{7}$$

The appeal of δ arises from being invariant under rotation. When $\delta < 0$, the degenerate point has three hyperbolic sectors. The pattern of eigenvector fields corresponding to the *trisector point* is shown in Figure 1. When $\delta > 0$,the degenerate point has one

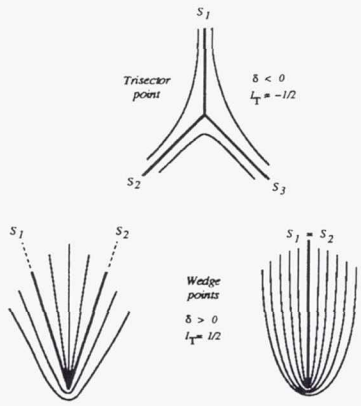


Figure 1: Trisector ($\delta < 0$) and wedge ($\delta > 0$) points.

hyperbolic sector. The local pattern corresponds to the *wedge point* represented in Figure 1.

3-D Tensor Fields

The eigenvalues of a 3-D tensor $\mathbf{T}$ can be obtained by investigating its characteristic equation.

$$\begin{vmatrix} \lambda - T_{11} & -T_{12} & -T_{13} \\ -T_{12} & \lambda - T_{22} & -T_{23} \\ -T_{13} & -T_{23} & \lambda - T_{33} \end{vmatrix} = \lambda^3 + P\lambda^2 + Q\lambda + R \quad (8)$$

where:

$$P = T_{11} + T_{22} + T_{33} \quad (9)$$

$$Q = \begin{vmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{13} \\ T_{13} & T_{33} \end{vmatrix} + \begin{vmatrix} T_{22} & T_{23} \\ T_{23} & T_{33} \end{vmatrix} \quad (10)$$

$$R = \begin{vmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{vmatrix} \quad (11)$$

Similar to critical point analysis, one can see that the coefficients P, Q and R are all tensor invariants.

In P-Q-R space, it can be shown that the solution trajectories only exist between surfaces S_1 and S_2, which are, respectively, given by:

$$\frac{2P^3 + 9PQ + 2\left(P^2 + 3Q\right)^{3/2}}{27} + R = 0 \quad (12)$$

and

$$\frac{2P^3 + 9PQ - 2\left(P^2 + 3Q\right)^{3/2}}{27} + R = 0 \quad (13)$$

On surface S_1, $\lambda_1(\overline{x}_0) = \lambda_2(\overline{x}_0)$; and on surface S_2, $\lambda_2(\overline{x}_0) = \lambda_3(\overline{x}_0)$. Triple degeneracy $\lambda_1(\overline{x}_0) = \lambda_2(\overline{x}_0) = \lambda_3(\overline{x}_0)$ occurs at points where S_1 and S_2 meet, i.e. $R = \frac{P^3}{27}$, $Q = \frac{P^2}{3}$.

Since the tensor and its deviator have the same topology, it is sufficient to examine only the deviator $\mathbf{D}$. By definition, $D_{11} + D_{22} + D_{33} = 0$, and therefore, $P = 0$. The coefficient Q can also be presented as:

$$Q' = -\frac{1}{2}\left(D_{11}^2 + D_{22}^2 + D_{33}^2 + 2D_{12}^2 + 2D_{13}^2 + 2D_{23}^2\right) \quad (14)$$

The characteristic equation now becomes:

$$\lambda^3 + Q'\lambda + R' = 0 \quad (15)$$

The P-Q-R space now reduces to the Q'-R' plane. On this plane, the solution trajectories exist between curves L_1 and L_2 (shaded area). L_1 and L_2 are given respectively by

$$\frac{R'}{2} = \left(\frac{Q'}{3}\right)^{\frac{3}{2}} \quad (16)$$

and

$$\frac{R'}{2} = -\left(\frac{Q'}{3}\right)^{\frac{3}{2}} \quad (17)$$

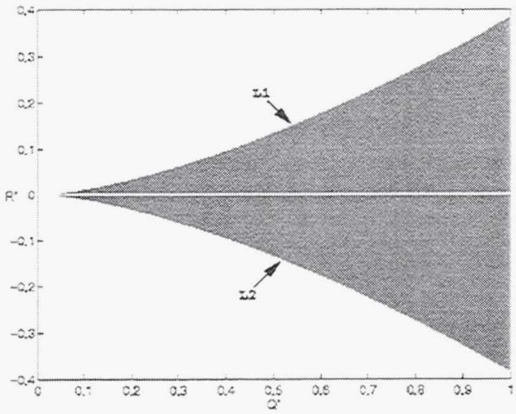


Figure 2: Q'-R' plane

Double degeneracy occur on the curves L_1 and L_2 and triple degeneracy occurs where L_1 and L_2 meet, i.e. $Q' = R' = 0$.

For double degeneracy cases, the pattern of the hyperstreamlines resembles that of a 2-D tensor field. For example, consider a degenerate point where $\lambda_1(\overline{x}_0) = \lambda_2(\overline{x}_0) > \lambda_3(\overline{x}_0)$. The tensor field is degenerate in the plane orthogonal to $\overline{v}_3(\overline{x}_0)$ within which locally two-dimensional patterns such as wedge points and trisectors can occur.

However, the case of triple degeneracy is global and the structure of the eigenvector trajectories in the vicinity of the degenerate point is fully three dimensional. The general structure of the eigenvectors in the vicinity of a point of triple degeneracy is based upon the fundamental 2-D degeneracies in planes defined by the 3-D tensor and its expansion in the neighborhood of $(\overline{x}_0)$.[17]

Case Studies

2-D flow past a hemisphere cylinder

The concepts discussed above are illustrated by first visualizing the topology of the stress tensor of a 2-D flow past a cylinder. Fluid elements undergo compressive stresses while moving with the flow. Stresses are described mathematically by the stress tensor, which combine isotropic pressure and anisotropic viscous stresses (deviator). Both eigenvalues of the stress tensor are negative and the two orthogonal eigenvectors, $\overline{v_1}$ and $\overline{v_2}$, are along the least and the most compressive directions, respectively. At a degenerate point, the viscous stresses vanish and both eigenvalues are equal to the pressure; degenerate points are points of pure pressure. In Figure 3, the texture[‖] represents the most compressive eigenvector of the stress tensor $(\overline{v_2})$. Color encodes the magnitude of the compressive force (λ), from most compressive(red) to least compressive(blue). Red dots marked with **W** are wedge points; white dots marked with **T** are trisectors. [**]

[‖]The texture is created by a technique discussed in References.[16]

[**]This part of the work has been done by Thierry Delmarcelle, for details refer to [16]

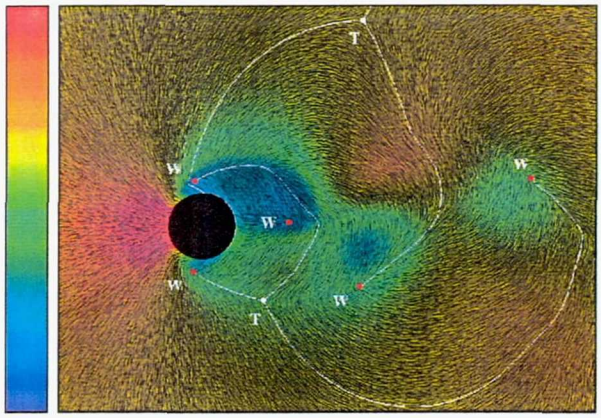


Figure 3: Stress in a flow past a cylinder; most compressive field.

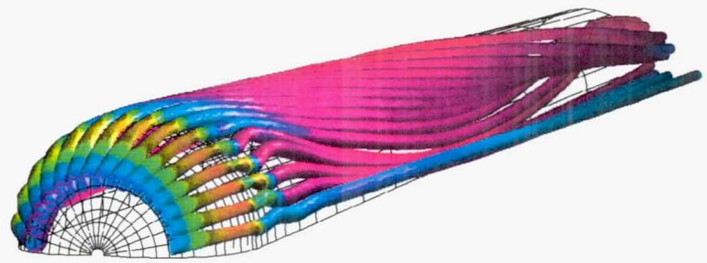


Figure 4: Reversible momentum flux density tensor in the flow past a hemisphere cylinder

Vortical Flow about a Slender body of Revolution

In the next example, hyperstreamlines are used to correlate several different physical quantities in fluid flow visualization. For the reversible part of the momentum-flux-density tensor $\Pi^r_{ij} = p\delta_{ij} + \rho v_i v_j$ (ρ is the mass density and v_i and v_j are the velocity components in the i, j direction, respectively), one may correlate pressure p, velocity direction, and kinetic energy density $\frac{1}{2}\rho v^2$ (ρ is the velocity magnitude). Indeed, the tensor field Π^r can be diagonalized as

$$\Pi^r = \begin{pmatrix} p + \rho v^2 & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix}$$

The major eigenvalue of Π^r is $\lambda_1 = p + \rho v^2$ and the associated unit eigenvector is aligned with the velocity vector. The other eigenvalues are degenerate and equal to the pressure ($\lambda_2 = \lambda_3 = p$) in the whole space. It follows that only major eigenvectors can be traced. Their trajectories are tangent to the velocity field and correspond to streamlines of the velocity field. The tubes' cross-sections are circular with a diameter proportional to the pressure p.[16] In Figure 4, color encodes the kinetic energy density. The direction of the incoming flow is 5° with respect to hemisphere axis, the Reynolds number is 14,000 and the flow is incompressible. The detachment at the end of the cylinder is clearly visible. The pattern of hyperstreamlines indicates that momentum is transferred from the nose of the body to the end fairly uniformly with a globally decreasing kinetic energy, as shown by color variations. However, there is a sudden change of kinetic energy (color) and pressure (diameter) associated with a significant variation of

the direction of the first five tubes.

As noted before, a tensor field and its deviator share the same topological structure. The importance of a deviator in tensor analysis can be demonstrated by the following discussion and illustration of the deformation tensor in a stationary flow field.

For an incompressible stationary flow the deformation tensor, defined as $\mathbf{Def} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}$, has a zero isotropic part and therefore is equal to its deviator. For a compressible flow, the deformation tensor is composed of a deviator superimposed on a non-zero isotropic tensor which represents the rate of expansion. Therefore, a deviator describes the topological structure for both incompressible and compressible stationary flows.

Following the assumption that for a rotational flow, inside the vortex core, the flow is purely rotational. Assume that the flow advances in the z-direction and rotates about the z-axis while the velocity within the vortex core area is $(-\omega y, \omega x, v_z)$ (where ω is the angular velocity and v_z is the axial velocity and is a constant for a stationary flow) and its deformation tensor $\mathbf{Def}\,(r < R)$ becomes singular; outside the vortex core, the velocity is $\frac{\Gamma}{r^2}(-y, x, 0)$ and its deformation tensor $\mathbf{Def}\,(r > R)$ is:

$$\frac{2\Gamma}{r^2}\begin{pmatrix} xy & -x^2+y^2 & 0 \\ -x^2+y^2 & -xy & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{18}$$

Here r is the distance from a point to the center of the vortex core and R is the radius of the vortex core. It is virtually a 2-D tensor with major and minor eigenvalues having equal magnitude but opposite sign and the medium eigenvalue remains zero. The deformation tensor is discontinuous at $r = R$. The angles of separatrices can be calculated by using the tensor in the neighborhood of the vortex core $\mathbf{Def}\,(r > R)$.[16] However, there is no real solution for the angles. This indicates that the major and minor eigenvector fields are a pair of loci in the transverse plane while the medium eigenvector follows the direction of the vortex core.

Figure 5 shows hyperstreamlines of the deformation tensor of flow past a hemisphere-cylinder. The angle of attack is set at 19° with a Reynolds number of 445,000 and a freestream Mach number of 1.2. There is a pair of primary vortices in this flow. The eigenvector field around each core forms a focus swirling opposite to each other. Both minor and major eigenvector fields form a ring pattern. The upper ring in Figure 5 is a minor hyperstreamline and the ring in the lower part which encloses the body is the major hyperstreamline. The two hyperstreamlines along the body are the

Figure 5: Deformation tensor in a flow past a hemisphere cylinder at incidence

medium eigenvectors and also define the direction of the vortex core.

Near Field of a Wing-Tip Vortex

Eddie viscosity turbulence models have been widely used in the simulation of turbulent flow, mainly for modeling attached flows. Under certain conditions, some of the models can be modified and extended to model separated flows around slender bodies of revolution at high angle of incidence.

In the following example, the near field of a wingtip vortex is examined to determine whether the basic assumptions behind the eddie viscosity turbulence models are valid. If not, the question as to why do they still assist in simulating the flow with relatively reasonable results needs to be answered. In an experiment that was conducted by Chow and Zilliac, a

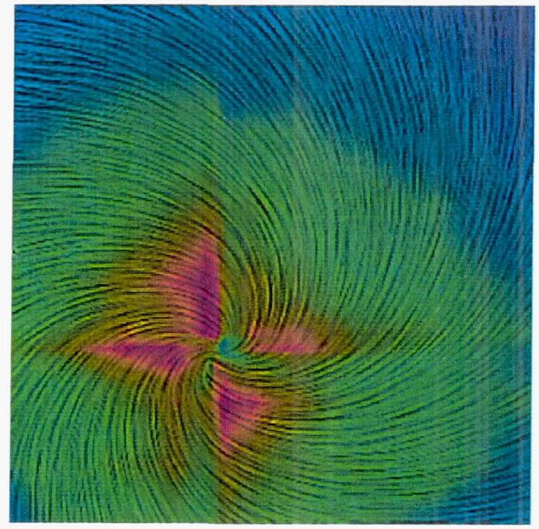


Figure 6: Deformation tensor in the near field of a flow past a wingtip; major eigenvector field

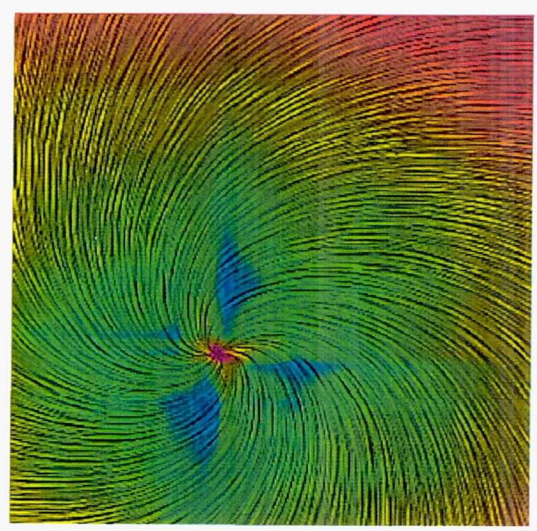


Figure 7: Deformation tensor in the near field of a flow past a wingtip; minor eigenvector field

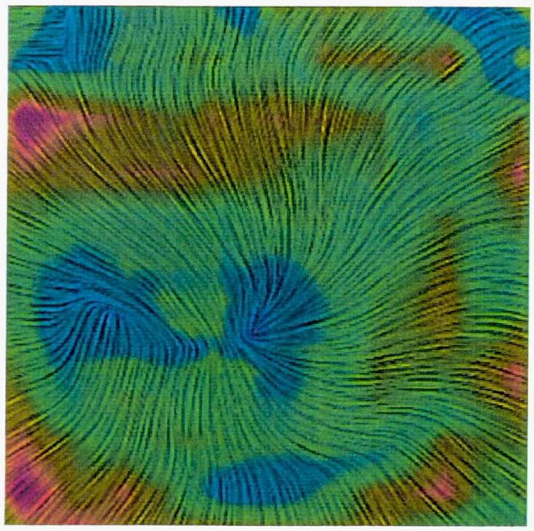


Figure 8: Reynolds stress tensor in the near field of a flow past a wingtip; major eigenvector field

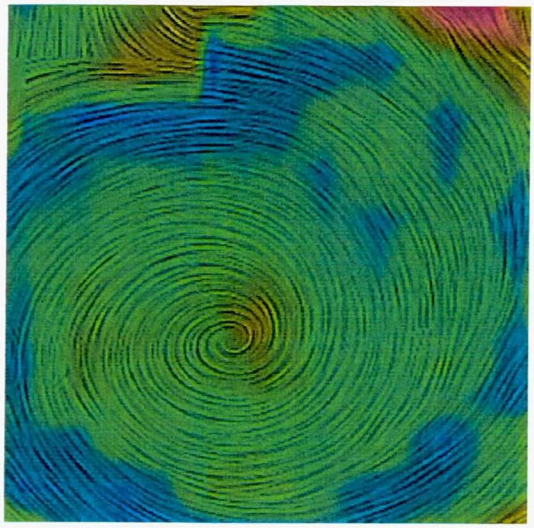


Figure 9: Reynolds stress tensor in the near field of a flow past a wingtip; minor eigenvector field

wide array of measurements, including surface oil-flow visualization, laser-illuminated smoke visualization, surface pressure and velocity-field measurements by use of a 7-hole pressure-probe and triple-wire hot-wire anemometry, was completed for the flow over a rectangular wing with rounded tip, up to 0.67 chords downstream of the trailing edge. The angle of attack of the wing was set at 10° and a chord Reynolds number was chosen at 4.6 million based on the desire to study a fully turbulent vortex. [††]

The primary objective of post measurement analysis is to compare the deformation tensor with the Reynolds stress tensor. Topological methods are ideally suited for this purpose. Previously, methods for visualization of tensor fields ranged from comparing the individual components of the tensors to comparing the eigenvalues and eigenvectors. However, these conventional methods have failed to determine whether the tensors are similar. Here we describe a new approach.

The eigensystems are computed for both Reynolds stress and deformation tensors obtained from experimental data. Texture maps are used to display the fields in the transverse plane across the vortex core.

[††] for detailed information about this experiment, please see[18]

From the eddy-viscosity hypothesis,

$$-\rho R_{ij} = \rho \nu_T \frac{d\bar{U}_i}{dx_j} \tag{19}$$

Where R_{ij} is the Reynolds stress, $\frac{d\bar{U}_i}{dx_j}$ is the mean deformation tensor and ν_T is the kinematic eddy viscosity. The eigenvalues of the Reynolds stress tensor are equal to the negatively scaled eigenvalues of the mean deformation tensor. Therefore, the order of eigenvalues of two tensors are opposite to each other, i.e. $\lambda_{R1} > \lambda_{R2} > \lambda_{R3}$ while $\lambda_{U1} < \lambda_{U2} < \lambda_{U3}$. This leads to the fact that the major eigenvector field of the mean deformation tensor swirls in the same direction as the minor eigenvector field of the Reynolds stress tensor. Figure 6 and Figure 7 display major and minor eigenvector fields for the deformation tensor, respectively. Their patterns show a pair of loci which confirms the discussion above regarding the behavior of eigenvector fields of a deformation tensor in a vortex core area. Figure 8 and Figure 9 are major and minor eigenvector fields for the Reynolds stress tensor, respectively. Their patterns also show a pair of loci. However, the minor eigenvector swirls faster than that of the deformation tensor, and therefore has a tighter focus. Similarly, the major eigenvector swirls slower and has a looser focus than the minor eigenvector fields of the deformation tensor. The similarity of the eigenvector fields between the deformation tensor and the Reynolds stress shows that the basic assumption behind the eddie-viscosity turbulence model are valid to a limited extent because the fields of these two tensors have different swirling speeds. This indicates that the tensors are not aligned and therefore the model is too simple for capturing the complicated turbulent flow in the near field of the wingtip. It is suggested that the model can be improved by taking into account the variation in swirling speed.

Acknowledgment

We thank Greg Zilliac of NASA Ames Research Center for providing the data and helpful discussions.

The authors are supported by NASA under contract NAG 2-911 which includes support from the NASA Ames Numerical Aerodynamics Simulation Program and the NASA Ames Fluid Dynamics Division, and also by NSF under grant ECS-9215145.

References

[1] T. Delmarcelle and L. Hesselink, "Visualization of second-order tensor fields and matrix data," in *Proc. IEEE Visualization '92*, pp. 316–323, CS Press, Los Alamitos, CA., 1992.

[2] M. Henle, *A Combinatorial Introduction To Topology.* Dover Publications, Inc., New York, 1994.

[3] J. Jimenez, "Kinematic alignment effects in turbulent flows," *Physics of Fluids*, pp. 652–654, 1992.

[4] R. K. Wm.T. Ashurst, A.R. Kerstein and C. Gibson, "Alignment of vorticity and scalar gradient with strain rate in simulated navier-stokes turbulence," *Physics of Fluids.A*, pp. 2343–2353, 1987.

[5] W. Kaplan, *Ordinary Differential Equations.* Addison-Wesley, REading, MA, 1958.

[6] L. Pontryagin, *Ordinary Differential Equations.* Addison-Wesley, REading, MA, 1962.

[7] A. V. A.A. Andronov and S. Khaikin, *Theory of Osillators.* Pergamon, Oxford, 1966.

[8] N. Minorsky, *Nonlinear Osillators.* Van Norstrand, Princeton, NJ, 1962.

[9] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *Computer*, vol. 22, pp. 27–36, Aug. 1989. Also appears in *Visualization in Scientific Computing*, G. M. Nielson & B. Shriver, eds. Companion videotape available from IEEE Computer Society Press.

[10] J. L. Helman and L. Hesselink, "Visualization of vector field topology in fluid flows," *IEEE Computer Graphics and Applications*, vol. 11, no. 3, pp. 36–46, 1991.

[11] A. P. M. Chong and B. Cantwell, "A general classification of three-dimensional flow fields," *Physics of Fluids A*, pp. 765–777, 1990.

[12] Y. L. Yingmei Lavin and L. Hesselink, "Singularities in nonuniform tensor fields," in *IEEE Visualization '97 Proc.*, CS Press, Los Alamitos, CA., 1997.

[13] A. I. Borisenko and I. E. Tarapov, *Vector and Tensor Analysis with Applications.* Dover Publications, New York, 1979.

[14] T. Delmarcelle and L. Hesselink, "Visualizing second-order tensor fields with hyperstreamlines," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25–33, 1993.

[15] T. Delmarcelle and L. Hesselink, "A unified framework for flow visualization," in *Computer Visualization* (R. Gallagher, ed.), ch. 5, CRC Press, 1994.

[16] T. Delmarcelle, *The Visualization of Second-Order Tensor Fields.* PhD thesis, Stanford University, 1994.

[17] Y. L. Yingmei Lavin and L. Hesselink, "The topology of three-dimensional symmetric tensor fields," in *Late Breaking Hot Topics IEEE Visualization '96*, pp. 43–46, CS Press, Los Alamitos, CA., 1996.

[18] J. D.-M. G.G. Zilliac, J.S. Chow and P. Bradshaw, "Turbulent structure of a wingtip vortex in the near field," *AIAA 24th Fluid Dynamics Conference*, 1993.

Summary of Work on Shock Wave Feature Extraction in 3-D Datasets

Principal Investigator: Lambertus Hesselink*
Stanford University, Stanford, California 94305-4035
Phone: (415)723-4850 Email: bert@kaos.stanford.edu

Abstract

This report discusses a method for extracting and visualizing shock waves from three dimensional computational data-sets. Issues concerning computation time, robustness to numerical perturbations, and noise introduction are considered and compared with other methods. Finally, results using this method are discussed.

Introduction

The visualization of shock waves within transonic, supersonic, and hypersonic flows has been a persistent problem. The difficulty lies in accurately extracting the shock from a computed data-set and presenting a meaningful representation of the results. Accurately extracting and depicting these shocks can lead to improvements in aerodynamic design and an insight into the physics of the flow. Much work has been done on the extraction of features within large data-sets using various data visualization techniques such as isosurfaces, stream ribbons, and contour maps. Recently, work done by Pagendarm and Seitz [1] has been successfully applied towards the extraction of shock waves. Their technique relied on the fact that the density jump across a shock numerically smears across a narrow band. This can be seen in Figure 1a, which represents a 1-D example, where the ξ axis is a location along the shock and ρ is the density. Since $\Delta\xi$ is a rather small distance, the shock can be assumed to be located at the inflection point. This point can be located via the second derivative, $\frac{\delta^2\rho}{\delta\xi^2} = 0$. Figure 1c depicts this derivative. Many numerical algorithms can be applied to extract this zero value. Pagendarm, and Seitz did not use the second derivative directly, since they were dealing with 3-D datasets; instead, they used $\vec{u} \cdot \nabla(\vec{u} \cdot \nabla\rho)$. There are a few difficulties with this method. From a numerical method standpoint, the noise is amplified for each derivative taken. Furthermore, this method requires the gradient to be taken twice, thereby producing many local minima and maxima which are detected by zero-search algorithms. These artifacts can be filtered with some computational effort. Another problem is that this method may cause false detection in free stream regions. The minute perturbations in numerical data in the free stream produce regions where the first and second derivatives

*Professor, Electrical Engineering and Aeronautics and Astronautics Departments.

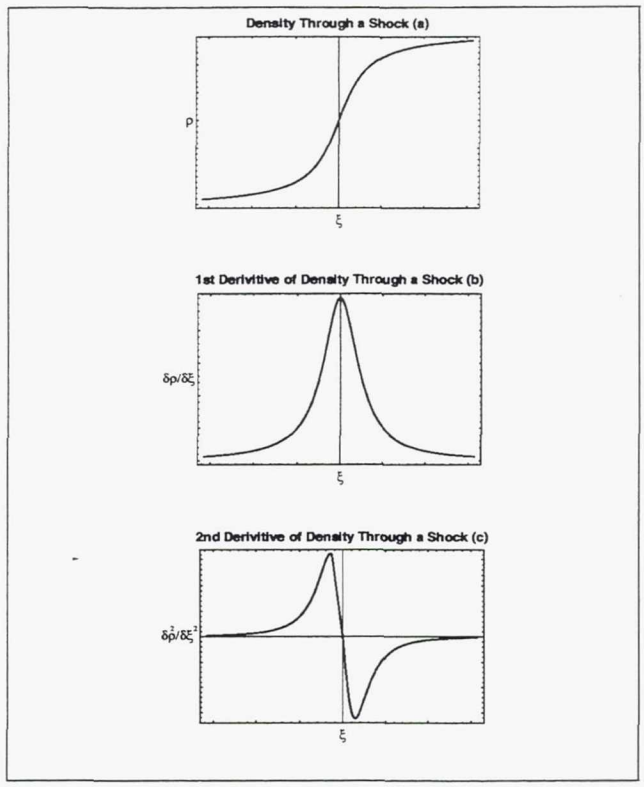


Figure 1: Density variation and its derivatives across a shock wave

appear to cross zero, rather than remaining fixed at zero. This causes zero-search algorithms to tag these points. Pagendarm's and Seitz's algorithm, however, is general enough to be applied to a variety of problems concerning discontinuities. What we propose is a technique that takes advantage of shock attributes for a faster and cleaner extraction. Faster in that we minimize the operation count, and cleaner in that we reduce noise.

Theoretical Background

A shock represents a sudden change of fluid properties. Typically, a shock is witnessed when a body travels at supersonic speeds. The flow adjusts to a body by abruptly changing its pressure, density, and temperature. This abruptness is caused by the flow's inability to sense the body. The Mach number M which is defined as the ratio of stream velocity to sonic velocity can be composed of two components namely one that is parallel $M_{||}$, and the other that is perpendicular $M_{\perp}$ to a shock wave. The fluid properties such as density, static pressure, and total temperature are distinct on either side of the shock and appear as a jump across the shock. We define $M_{\perp}$ as follows:

$$M_{\perp} = \frac{V_{\perp} \cdot V_{\perp}}{a^2} \tag{1}$$

where a is the speed of sound and $V_{\perp}$ is the velocity in the direction of the pressure gradient,

$$V_{\perp} = V \cdot \frac{\nabla P}{|\nabla P|} \tag{2}$$

The transition of $M_\perp$ across a shock progresses from supersonic, $M_\perp > 1$ to subsonic, $M_\perp < 1$. Equation (1) assumes that the pressure gradient is normal to the shock. We use this assumption in our method for shock extraction.

Several steps are required for the extraction of the shock. We first begin by computing $M_\perp$ for the entire computational space using Equation (1). $M_\perp$ is used, rather than Mach number, to ensure the capture of oblique shock waves. The computational space is composed of grid cells. Each cell, composed of eight neighboring grid points, is checked to ensure the region is in compression $\nabla P \cdot \vec{V} > 0$. Shocks exists only in regions of compression; by removing expansion waves, further computation is not required. A further check to validate the existence of a shock within a cell, is to validate that the change in pressure (δP) is within a safety factor of the Rankine-Hugoniot range. Regions where the pressure gradient changes direction abruptly, thereby producing a large angle between the velocity vectors, can cause $M_\perp$ to drop below 1.0, and appear as a shock. Therefore, using the isentropic relation,

$$\delta P = P * \frac{2 * \gamma}{\gamma + 1} * M_\perp^2 - \frac{\gamma - 1}{\gamma + 1} \tag{3}$$

where $M_\perp$ is the largest in the computational set, we ensure that the pressure gradient is large enough for a shock to exist. If a cell meets these criteria, it is then passed through the marching cube algorithm [2]. Surfaces constructed from triangles of $M_\perp = 1.0$ are created. These triangles are further checked to ensure that the pressure gradient is in fact in the direction of the surface normal. This filters out regions near the body (in the boundary layer) that can be tagged as a shock since the velocity progresses from zero at the body (no slip condition) to free stream velocity, hence passing through $M_\perp = 1.0$. This method of shock detection requires only the first derivative ∇P to be computed, which introduces less noise and is computationally faster than other methods requiring a second derivative. Computation time is further reduced by making simple checks to ensure a region's candidacy for a shock prior to extensive processing. This method is also useful for finding shocks in solutions that have not converged fully. Free stream perturbations will not be tagged since characteristics that are indigenous to flow fields are used. Therefore, this method is not ideal for general 3-D data-sets such as medical imaging data but works best for flow problems.

Results

Several test cases were run to ensure the validity of the method. Figure 2 depicts a lamda shock extracted from a data-set of an ONERA M6 wing [1] with an incoming Mach number, $M_\infty = 0.84$, angle of attack, $\alpha = 6^o$, and Reynolds number, $Re = 761,000$. As can be seen from the figure, we have a clean extraction of the shock with little to no artifacts. The extraction also reveals a region where the lamda shock is separated. This space informs the designer that the solution may not have converged fully, or a higher grid resolution is required in the area. As another verification of the shock algorithm, we extract shock features from a hemisphere cylinder at an angle of incidence of $\alpha = 19^o$, $M_\infty = 1.2$, and $Re = 445,000$). Figure 3 depicts the results. Two distinct shocks exist. The first is the bow shock which is upstream of the nose. The coarseness of the grid yields "holes" in the shock surface. Further

[1] ONERA (Office National D'etuedes Et De Recherches Aerospatiales) wing designed for studies of 3-D flows of low to transonic speeds at high Reynolds numbers.

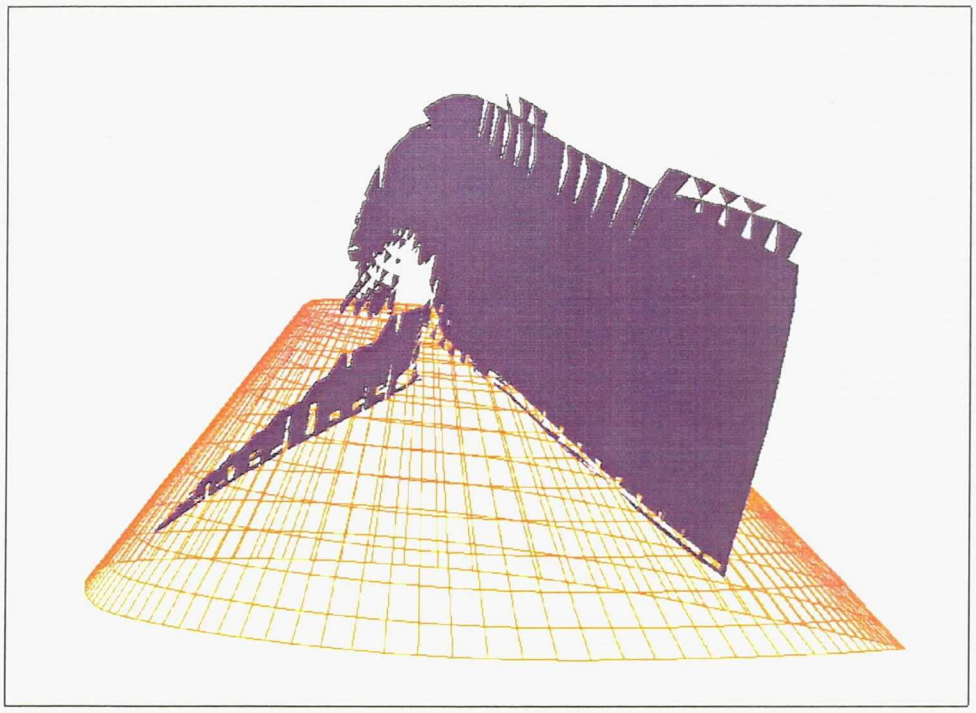


Figure 2: Lamda Shock extracted from ONERA wing.

downstream, along the top surface of the cylinder another shock forms due to the high angle of attack which causes the flow to reach supersonic speed on the leeward side of the body.

Summary

We have devised a method for shock extraction that uses knowledge of the flow to reduce computation time and noise. The basic steps of the method are as follows:

1. Calculate the Mach component in the direction of the pressure gradient ($M_\perp$).
2. Verify that the region is undergoing compression.
3. Verify that the pressure gradient within a region is consistent with the existence of a shock.
4. Use the marching cube algorithm to search for $M_\perp = 1.0$.
5. Disregard all surfaces whose normal is not in the same direction as the pressure gradient.

Future Work

We plan on extending this work to other significant flow features. The extraction of vortices within a flow will be the first area of study followed by viewing skin friction along a body. The skin friction will be depicted through a simulation oil flow combined with pressure sensitive paint.

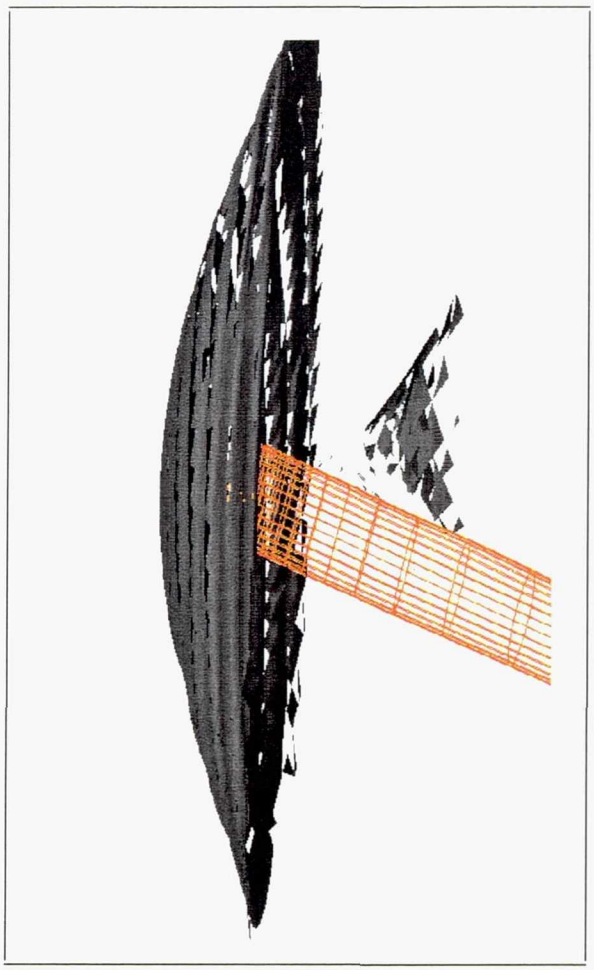


Figure 3: Bow shock with second shock on a hemi-spherical cylinder

References

[1] B. H.-G.Pagendarm, "An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves," in *Scientfic Visualization-Advanced Techniques* (P.Palamidese, ed.), ch. 4, Ellis Horwood Ltd., 1993.

[2] H. C. W. Lorensen, "A high resolution 3d surface construction algorithm.," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

Flow Visualization with Textures

Thomas Loser[1], Dieter Mewes[1], Yuval Levy[2], and Lambertus Hesselink[3]

[1] Institut für Verfahrenstechnik, Universität Hannover, Callinstr. 36, 30169 Hannover, Germany, {loser,dms}@c36.uni-hannover.de
[2] Faculty of Aerospace Engineering, Technion city, Haifa 32000, Israel, yuval@bellini.technion.ac.il
[3] Department of Electrical Engineering and Aeronautics, Astronautics, Stanford University, Stanford, California 94305-4035, USA, bert@kaos.stanford.edu

Abstract. The increasing usage of computers in experimental and theoretical research has presented new challenges to the scientific visualization community. This is especially valid for the case of multi-dimensional datasets as vector fields which are produced in the field of chemical engineering when flow and concentration fields or heat transfer are calculated or measured. Vector fields can be visualized in detail using the "Line Integral Convolution"(LIC) Algorithm from Cabral and Leedom [1]. The focus of this paper is a discusion of basic principles and features of LIC, its usage on curvilinear grids and a comparison to traditional visualization techniques as vector arrows and streamlines.

1 Introduction

Traditional approaches for flow visualization are symbolic representations such as streamlines or vector arrows to visualize the direction of a flow field. In recent years several new techniques for vector field visualization have been introduced among the visualization with textures appears to be one of the most promising. There are several definitions for textures in literature. Commonly Textures for flow visualization purposes are defined as *surface structures containing directional information.* An example of a texture is shown in Fig. 1. For the first time van Wijk [2] used a texture to visualize the flow on a ship hull. With the introduction of the "Line Integral Convolution"-algorithm by Cabral and Leedom in 1993 [1] the generation of textures became much easier. Since 1993 there are published a number of papers concerning the application and extensions of the LIC-algorithm. Forssell and Cohen [3] extended the LIC-algorithm to its application on curvilinear grids in 3D. As textures are by nature bidirectional, Delmarcelle and Hesselink [4,5] applied textures for the visualization of eigenvector fields of tensor fields. In order to visualize additionally the forward and backward direction and the velocity of a flow animation of textures has been introduced by Cabral and Leedom [1] and extended by several other research groups [3,6–9]. The generation of textures requires an input noise (cf. section 2). With a local variation of the frequency of the input noise Kiu and Banks [10] achieved special visual effects

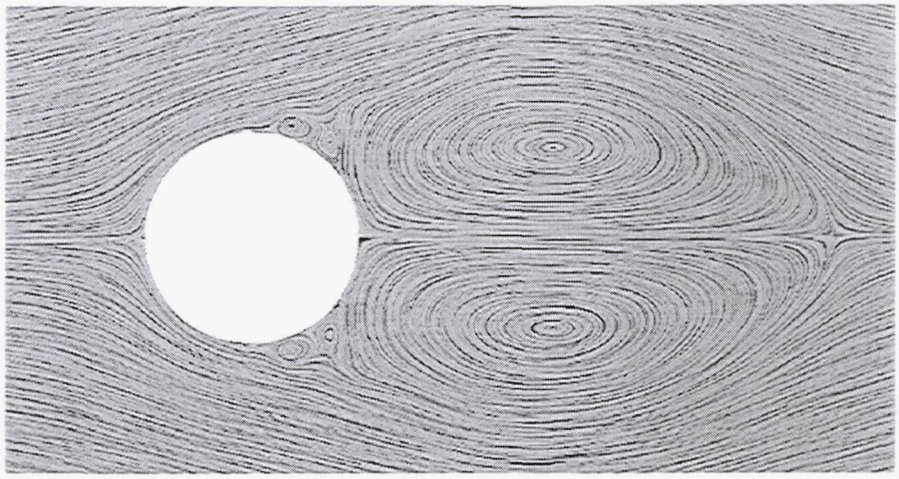


Fig. 1. Flow field past a zylinder visualized with a texture

using multi-frequency noise. Battke, Stalling and Hege [11] introduced an algorithm to calculate textures on arbitrary surfaces in 3D. In order to reduce the cost of computation for the calculation of textures, Stalling and Hege [6] introduced the algorithm "fastLIC", which allows a more efficient calculation of textures. The visualization of unsteady flow fields with textures has been investigated by Forssell and Cohen [3] and by Shen and Kao [12].

2 Generation of Textures with Line Integral Convolution

The basic principle of texture generation with the LIC-algorithm is to smear an input noise along the integral lines of a vector field. The input noise consists of random grey values for each pixel of the later generated texture. The smearing of the grey values can mathematically be expressed as a convolution of the grey values along the integral lines of the vector field. If I_{in} is the intensity of the input noise (grey values) and $\sigma(s)$ ist the path or integral curve

$$\frac{\partial \sigma(s)}{\partial s} = \frac{v}{|v|}, \tag{1}$$

where the tangent of σ has the direction of the vector field, then the intensity (grey) values I_{out} of a texture can be calculated with the equation

$$I_{out}(x,y) = \int_{s_0-L}^{s_0+L} I_{in}(\sigma(s))h(s-s_0)ds, \text{ with } (x,y) = \sigma(s_0). \tag{2}$$

In (2) h is the convolution filter function. This function has a strong influence on the appearance of a texture. The simplest filter function is a box filter with

$$h(s) = \begin{cases} 1 & \text{if } -L \leq s \leq L \\ 0 & \text{if } -L > s \text{ or } L < s, \end{cases} \tag{3}$$

where L is the filter length in each direction from the output pixel. Thus $l = 2 * L$ is the whole filter length. The longer the filter length, the longer the black and white streaks will appear in the texture. If a box filter is used for the convolution (2) can be simplified to the summation

$$I_{out}(x, y) = I_{out}(i, j) = \sum_{P \subset \tau(\sigma)} I_{in}(P) \Delta s, \tag{4}$$

where the intensity values I_{in} of the pixels are only weighted by the path Δs the streamline passes through this pixel. In (4) (i, j) are the pixel coordinates at (x, y) and P is a pixel of the set τ of all pixels which are intersecting with the path σ of the local streamline.

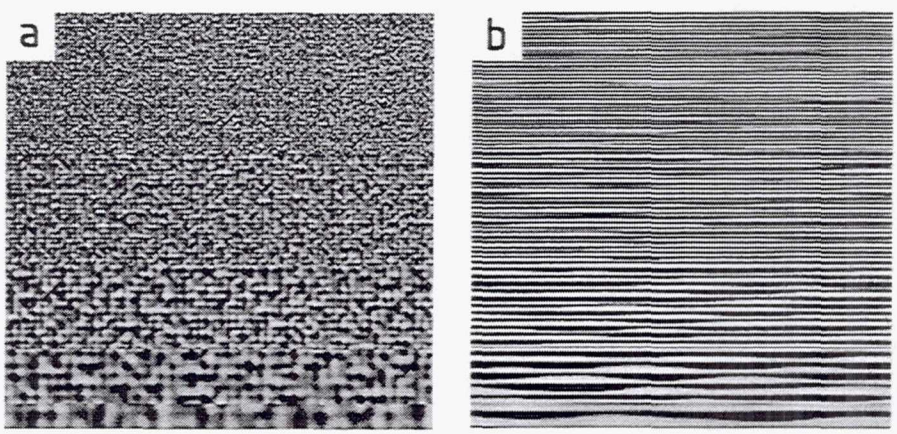


Fig. 2. Visualization of a boundary layer flow over a flat plate. (a) shows the multi-frequency input noise, (b) shows the generated texture

The whole procedure to generate textures can be subdivided into four steps:

1. Calculation of the input noise, which consists of random intensity (grey) levels in the area, where the texture has to be generated.
2. For each pixel a local streamline is calculated - starting at the center of this pixel in both directions.
3. The intensity values of the input noise are integrated along the local streamline within the distance $[-L, +L]$ and normalized by the integration length. The pixel values will be weighted by a convolution filter function.

4. The integration of the pixel values leads almost to an equalization of the intensity values. Therefore the contrast of the texture has to be increased.

The result of this procedure is a black and white texture as it is depicted in Fig. 1. After that the texture can be colored in order to visualize an additional dimension of information (cf. Sect. 3).

The type and frequeny of the input noise has a strong influence on the appearance of a texture. There are several algorithms to calculate the input noise [4,7]. The frequency of the noise can be locally varied in order to achieve an additional visual effect. In Fig. 2 the boundary layer flow on a flat plate is visualized using multi-frequency noise [10]. The velocity in the horizontal direction v_x is a function of the vertical distance y to the plate as $v_x \propto \sqrt{y}$. Besides the direction of the flow the texture in Fig. 2 gives an impression of the magitute of the velocity.

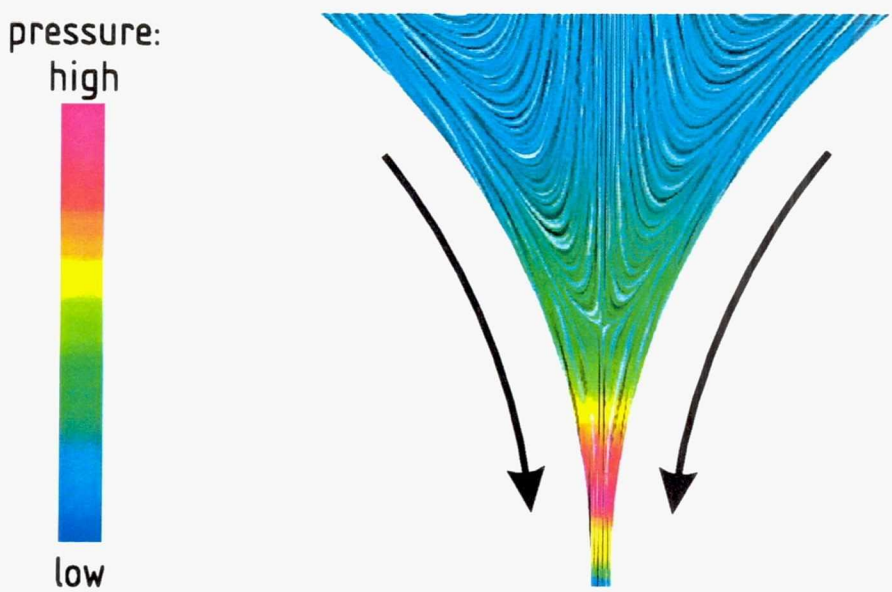


Fig. 3. Visualization of the flow field of a viscoelatic fluid between two rotating rolls. The color encodes the pressure. Image by courtesy of Karsten Riest [13]

3 Coloring Textures in Chemical Engineering

With colored textures it is possible to visualize one more dimension of information in the same image. The color can encode e.g. the magnitute of the velocity, the pressure, the volume fractions of the phases etc. In order to generate colored textures one first have to define a color map which gives for

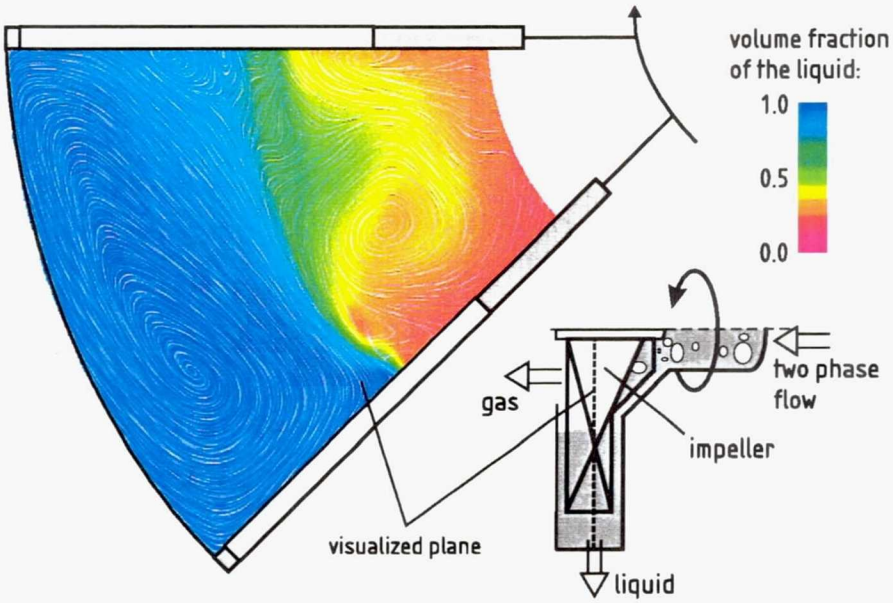


Fig. 4. Flow field in a two-phase separator. The color encodes the volume fraction of the liquid. Image by courtesy of Matthias Creutz [14]

each scalar value to be visualized a color, except black and white. Then the color values will be multiplied with the intensity value of the precalculated black and white texture [7]. In Fig. 3 the flow field between two rotating rolls is visualized where the color encodes the pressure. In such images the dependencies between pressure and flow topology can be studied easily.

In the example in Fig. 4 the flow field in a two-phase separator is depicted. The flow is caculated with a commercial CFD programm. With the color map the volume fraction of the liquid phase is encoded. The texture shows the results on the plane in the middle of one section of the impeller.

4 Textures on Curvilinear Surfaces

Many grids for numerical calculations are curvilinear grids. It is possible to generate LIC-textures on curvilinear surfaces of curvilinear grids [3,7]. In order to generate a texture on a curvilinear surface, first the vector field on this surface is mathematically transformed into a 2D surface. We denote this surface the computational space. The transformation into the computational space can be performed using the Jacobian matrix. In the computational space the texture will be generated according to the algorithm described above. The resulting 2D-texture in computational space will be mapped back to the curvilinear surface in 3D. This procedure is illustrated on the left side in Fig. 5. In order to generate a texture of the flow on the first slice of the

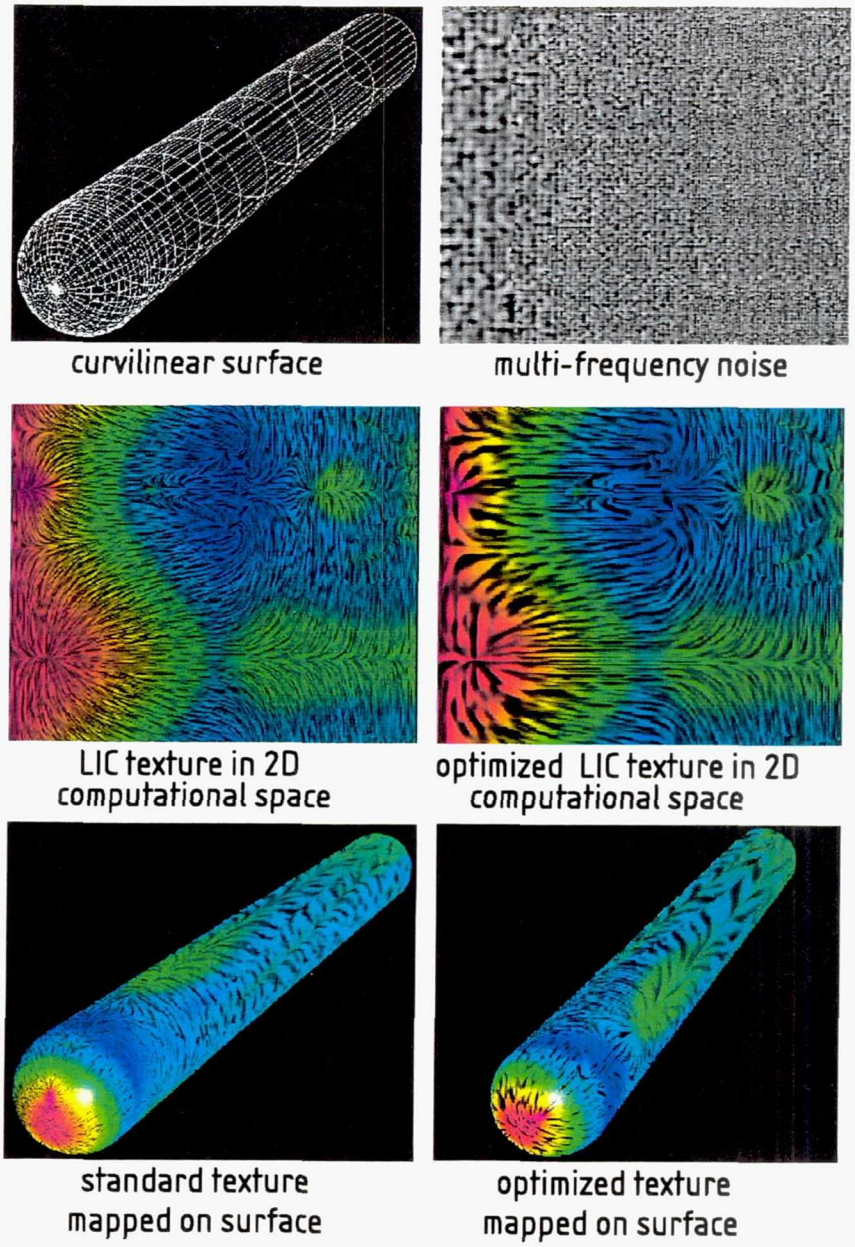


Fig. 5. LIC-textures on curvilinear grids. Left: Mapping a 2D texture on a curvilinear surface in 3D. Right: Optimize the streak size of the 3D-texture using multi-frequency noise

numerical grid of a hemishere cylinder (top), a LIC-texture in computational space is calculated (middle) and then mapped back to the curvilinear surface (bottom).

This procedure works well if the grid density is similar in all areas of the curvilinear surface. But for the hemisphere cylinder the grid density is much higher at the top of the hemishere than at the end of the zylinder. Thus the streaks which are homogeneous in the computational space vary significantly on the curvilinear surface in 3D (Fig. 5 bottom left). This can be compensated by using multi-frequency noise. The frequency of the noise is calculated high, where the grid density is low and vice versa (Fig. 5 top right). Thus locally different streak widths are calculated for the texture in computational space (Fig. 5 middle left). If this texture is mapped onto the curvilinear surface it results in an almost uniform streak width in 3D (Fig. 5 bottom right).

Table 1. Advantages and disadvantages of flow visualization with textures compared to traditional visualization techniques

visualization technique	advantages	disadvantages
vector arrows	simple depiction	details difficult to visualize
	direction clear	discretisation necessary
	length of arrow shows magnitude of velocity	visual clutter for high density of arrows
streamlines	visualization of flow path	details difficult to visualize
	quantitative comparisons of different flow fields possible	discretisation necessary, problems for convergence / divergence
		bidirectional
textures	visual appealing	high computational cost
	highest possible resolution of flow fields, details depicted	bidirectional
	good representation of flow topology	
	continuous representation	

5 Textures versus Traditional Visualization Techniques

The advantages and disadvantages of textures compared to traditional flow visualization techniques, among which vector arrows and streamlines are the

most common, are shown in Table 4. One of the major strength of textures is that with pixel-oriented textures the highest possible resolution of a flow field can be achieved. Thus even details like secondary vortices can be resolved and the flow topology is easy to extract graphically. The disadvantage of textures for flow visualization is their high computational cost and the bidirectional nature of textures. The later disadvantage can be overcome by animation of textures [6,7]. The bidirectional nature of textures makes them ideal for the visualization of bidirectional vector fields like eigenvector fields of tensor fields [5].

References

1. Cabral B., Leedom L.C. (1993) Imaging Vector Fields using Line Integral Convolution. In: Kajiya J.T. (Ed.) Computer Graphics (SIGGRAPH'93 Proceedings), 8, 27, 263-272
2. van Wijk J.J. (1991) Spot Noise. Texture Synthesis for Data Visualization. In: Computer Graphics (SIGGRAPH'91 Proceedings), 25, 4, 309-318
3. Forssell L.K., Cohen S.D. (1995) Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-Speed Animation, and Unsteady Flows. IEEE Transactions on Visualization and Computer Graphics, 1, 133-141
4. Delmarcelle T. (1994) The Visualization of Second-Order Tensor Fields. Ph.D. Thesis, Stanford University, Stanford, CA
5. Delmarcelle T., Hesselink L. (1994) The Topology of Symmetric, Second-Order Tensor Fields. In: Proceedings Visualization '94, IEEE Computer Society Press, Los Alamitos, CA, 140-147
6. Stalling D., Hege H.C. (1995) Fast and Resolution Independent Line Integral Convolution. In: Computer Graphics (SIGGRAPH'95 Proceedings), 249-256
7. Loser T. (1996) Entwickeln von Verfahren zur numerischen Strömungsvisualisierung, Diplomarbeit, Universität Hannover, Germany
8. Stalling D., Hege H.C. (1997) LIC: Acceleration, Animation, and Zoom. In: Texture Synthesis with Line Integral Convolution, SIGGRAPH 97, Course Notes 8, Los Angeles, 17-49
9. Jobard B., Lefer W. (1997) The Motion Map: Efficient Computation of Steady Flow Animations. In: Proceedings Visualization '97, IEEE Computer Society Press, Los Alamitos, CA, 323-328
10. Kiu M.H., Banks D.C. (1996) Multi-Frequency Noise for LIC. In: Proceedings Visualization '96, IEEE Computer Society Press, Los Alamitos, CA, 121-126
11. Battke H., Stalling D., Hege H.C. (1996) Line Integral Convolution for Arbitrary Surfaces in 3D. In: Polthier, Hege (Eds.) Visualization and Mathematics, Springer Verlag, Berlin Heidelberg
12. Shen H.W., Kao D.L. (1997) UFLIC: A Line Integral Convolution Algorithm for Visualizing Unsteady Flows. In: Proceedings Visualization '97, IEEE Computer Society Press, Los Alamitos, CA, 317-322
13. Riest K., Mewes D. (1998) Flow of Viscoelastic Fluids between two Rotating Rolls. In: Proceedings of 9th Int. Coating Sci. and Techn. Symp., Newark, DE, 259-262
14. Creutz M. (1998) Separation dreiphasiger instationär strömender Gemische. Ph.D. Thesis, Universität Hannover, VDI-Verlag, Düsseldorf